Identifying and Predicting Momentum Turning Points Using Fundamental Time-Series Data

HWYC4

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Master of Science in Computational Finance
of
University College London

Department of Computer Science

September 6, 2024

Declaration

I, candidate HWYC4, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Signed: HWYC4

Abstract

This thesis presents original research on developing a comprehensive methodology for predicting stock momentum turning points. These mark reversals in stock price trends, either from uptrend to downtrend, or vice versa[1]. The primary aim of this study is to classify momentum turning points using time-varying company fundamentals, with the premise that if a particular trend changes direction, we can identify and capitalise on this new trend as early as possible. Partnering with Banking Science Limited has been instrumental in providing essential data and support for this analysis, ensuring that this research is both relevant and applicable in real-world contexts.

The study is divided into the following sections:

1. *Exploratory Analysis and Momentum Driver Feature Engineering.* This chapter presents the first experiment, which describes a systematic methodology for engineering sector-specific fundamental drivers of momentum in stock markets. It begins with the acquisition of relevant data, followed by extensive preprocessing, including the filtration of the database, cleansing of the data and transformation of fundamentals to engineer features. This analysis forms the foundational basis for further investigations into stock momentum in this study.

2. *Finding Ideal Turning Point Labels.* This chapter details the second experiment, which focuses on producing turning point labels that will be used to train the classification models in the final experiment. The chapter begins by stating a concise definition for a momentum turning point. It then proceeds by describing two promising algorithms which can be used to label momentum turning points in the price data. These labels will be used for the models in experiment 3.

3. *Identifying Key Momentum Drivers and Predicting Momentum Turning Points.* This chapter focuses on the third and final experiment, involving the reduction of the feature space by sector and the development of classification models to predict stock momentum turning points ahead of time. The study leverages the features from experiment 1 and the labels from experiment 2. By employing hyperparameter tuning along with cross-validation, the study aims to reliably measure the out-of-sample performance of the different turning point classification models.

4. *Results.* This section presents the outcomes of the applied methodologies for finding the most important features and identifying and predicting momentum turning points. It details the performance of the different models at each stage of the pipeline, comparing their effectiveness in terms of accuracy and reliability through thorough analysis of suitable performance metrics.

This study makes the following contributions to science:

1. *Enhanced Feature Engineering and Selection.* The approach taken for engineering and selecting features using a large dataset is not only pivotal in financial analysis but

also extends to various STEM fields, demonstrating a versatile methodology that can be adapted for diverse analytical challenges across different domains.

2. *Innovative Turning Point Labelling Technique for Noisy Data.* The study's novel approach to labelling significant turning points can be applied to several domains where it is necessary to discern the trends and reversals in multiple different variables simultaneously.

3. *Advancements in Stock Price Prediction.* We showcase the use of machine learning (ML) techniques to predict different characteristics of stock price data. This enables a deeper understanding of financial market dynamics and reinforces the role of artificial intelligence in financial forecasting and decision-making.

Impact Statement

This thesis advances empirical research by innovatively using fundamental features to classify stock momentum. The research establishes a novel framework that directly contributes to the enhanced predictive accuracy of stock performance and momentum trading strategies:

1. *Optimised Data Exploration and Feature Selection for Enhanced Insights.* The approach to data exploration and feature selection in this study significantly refines the analysis process, allowing for the streamlined extraction of meaningful insights from complex datasets. By focusing on the most informative features, these algorithms not only optimise data processing, but also improve strategies for asset allocation and risk management.

2. *Precise Momentum Turning Points for Trading Strategy Optimisation.* The methodology for identifying precise turning points in asset momentum also helps in improving trading strategies by enabling the identification of optimal entry and exit points of different trading positions.

3. *Robust Predictive Methodologies for Proactive Portfolio Management.* The prediction methodology equips financial analysts with a robust decision-support tool that is vital for navigating complex market dynamics. The research facilitates proactive portfolio management by enabling timely responses to changes in market conditions, which is crucial for avoiding significant losses and capturing potential gains.

Contents

1 Introduction 1

1.1 Research Motivation . . . 2

1.2 Research Objectives . . . 3

1.3 Research Experiments . . . 3

1.4 Scientific Contributions . . . 4

1.5 Thesis Structure . . . 5

2 Background and Literature Review 7

2.1 Factor Investing and Fundamentals as Stock Price Predictors . . . 7

2.2 Machine Learning and Other Statistical Analyses . . . 9

2.2.1 Models . . . 9

2.2.2 Hyperparameter Tuning . . . 13

2.2.3 Model Evaluation . . . 13

2.3 Traditional Momentum Investing and Related Work . . . 16

2.3.1 Momentum Metrics and the Momentum Factor . . . 16

2.3.2 Momentum Turning Points . . . 18

3 Experiment 1 - Exploratory Analysis and Momentum Driver Feature Engineering 19

3.1 Data Preprocessing . . . 19

3.1.1 Database Filtering . . . 20

3.1.2 Adjusting the Price Data for Corporate Actions . . . 21

3.1.3 Data Cleaning . . . 23

3.2 Feature Engineering . . . 25

3.2.1 Approach 1: The Rate-of-Change Transformation . . . 25

3.2.2 Approach 2: Sector-Relative Feature Engineering . . . 26

3.3 Summary . . . 27

4 Experiment 2 - Finding Ideal Turning Point Labels 29

4.1 Defining a Series of Momentum Turning Points . . . 29

4.1.1 Price Turning Points vs Momentum Turning Points . . . 29

4.2 Labelling Momentum Turning Points 30
4.2.1 Deviation Algorithm Implementation 31
4.2.2 Max-Min Windowing Algorithm Implementation 33
4.3 Summary 35

5 Experiment 3 - Identifying Key Momentum Drivers and Predicting Turning Points 37
5.1 Practical Considerations 37
5.2 Feature Selection 38
5.2.1 Structuring the Data for Feature Selection 38
5.2.2 Removing Static and Low Variance Features 39
5.2.3 Finding the Most Important Features 39
5.3 Structuring the Data for Class Prediction 42
5.4 Predicting Turning Points 43
5.4.1 Calculating Class Weights 43
5.4.2 Cross-Validation and Evaluation 44
5.5 Summary 45

6 Results 47
6.1 Engineered Features 47
6.2 Turning Point Labels 49
6.2.1 Graphical Validation of the Max-Min Windowing Algorithm 50
6.3 Key Momentum Drivers 51
6.4 Class Predictions 53
6.4.1 Class Weights and Hyperparameter Optimisation 53
6.4.2 Out-of-Sample Evaluation 54
6.5 Summary 56

7 Conclusions and Future Work 58
7.1 Conclusions 58
7.2 Strengths and Limitations 59
7.3 Future Work 60
7.4 Final Comments 61

Bibliography 61

Appendix A Columns Removed and Retained During Preprocessing and Feature Selection 65
A.1 Non-Double-Type Columns Removed 65
A.2 Features Dropped Due to Missing Values Thresholding 66
A.3 Features Dropped Due to Low Variance 74

A.4 Top Quintile of Features from Random Forest Selection 75
A.5 Dropped Features from Correlation Analysis 76
A.6 Dropped Features from Mutual Information Analysis 77
A.7 Remaining Features - The Key Momentum Drivers 77
A.8 Feature Importance Graphs 79

Appendix B Code and Visualisations - Max-Min Windowing Algorithm 81

B.1 Code for Labelling Momentum Turning Points in Price Data 81
B.2 Momentum Turning Point Visualisations for Max-Min Windowing Approach 86

List of Figures

2.1 Visual Representations of the Random Forest Mechanism and the Iterative Error Reduction Process for GBMs 11

2.2 Illustration of Rolling-Forward Cross-Validation with Expanding Training Folds[2] 14

3.1 Historical Daily Prices tables and Quarterly Earnings Reports tables to be queried 21

3.2 Pie Chart of Stocks by Sector 21

3.3 Pipeline for Aligning Fundamental and Price Data 23

4.1 Illustration of Price Turning Points in Red and Momentum Turning Points in Green 30

4.2 Illustration of the Deviation Algorithm's Failure to Capture Significant Peaks and Troughs. The significant peak and trough have been manually labelled. 32

5.1 Random Forest Importances Before Correlated Feature Removal; Mutual Information Scores After Correlated Feature Removal - Finance Sector . . . 41

5.2 Comparison of Correlation Matrices Before and After Feature Selection - Finance Sector 42

5.3 Illustration of Conversion to Weekly Labels for a Single Stock. Each '...' represents intermediate days with 'Regular' labels. Each new label reflects the class of momentum turning point(s) over the following four weeks. Day γ denotes γ days after 08/03/24. 42

6.1 Distribution of Market Cap Categories in the Finance Sector 47

6.2 Maximum and Minimum Momentum Turning Points for Various Stocks . . 50

6.3 Summary of Feature Selection and Removal for the Finance Sector. Graphs (a), (b), (c) correspond to tables (d), (e), (f) respectively. 51

A.1 Top Quintile of Features Selected from RF Regression Against Returns . . . 79

A.2 Correlation Matrix Before Correlated Feature Removal 79

A.3 Correlation Matrix After Correlated Feature Removal 80

A.4 Aftermath of Correlation Analysis: MI Shared with Returns for Remaining Features 80

B.1 Daily Turning Point Labels for MS9FKZ-S 86

B.2 Daily Turning Point Labels for HVQ6TN-S 86

B.3 Daily Turning Point Labels for C0TNSZ-S 87

B.4 Daily Turning Point Labels for KHXMCD-S 87

List of Tables

2.1 Comparison of Class-Specific and Weighted Metrics 15

3.1 The 19 Significant Stock Market Sectors 20
3.2 Example price adjustment over time using split factors (SFs) 22
3.3 Illustration of Price Anomaly Detection and Correction for Single Stock . . 24
3.4 Illustration of Sector-Relative Feature Engineering for the Revenue Feature 26
3.5 Cleansing Stats for Finance Sector 27

4.1 Illustration of Boundaries and Initial Labels for Two Consecutive 40-Day Intervals for a Single Stock, where ”...” in the Label Column is a Series of ’Regular’ Labels 33
4.2 Illustration of Min/Max Label Replacement at Adjacent Interval Boundaries 34
4.3 Illustration of Min/Min Label Replacement at Adjacent Interval Boundaries 34
4.4 Illustration of Consecutive Maximum 0-1 Label Replacement 34
4.5 Illustration of Consecutive Maximum 0-0 Label Replacement 35
4.6 Illustration of Insignificant TPs Label Replacement (less than 5% price change in 4 weeks) 35

5.1 Optimised Hyperparameters for Random Forest Feature Importance Model 40
5.2 Train-Validation Splits for the Years 2000-2024. The first table covers the years 2000-2011, while the second table covers 2012-2024, with ‘X’ indicating years used for training, ‘O’ for validation, and ‘*’ for testing. 45

6.1 Filtered Dataframe Showing Engineered Feature Time-Series for ff_sales and ff_pbk for 3 Stocks from the Finance Sector from End of March to End of April 2000. Note that the ’fsym’ column contains the stock identifier. . . . 48
6.2 Distribution of Daily Turning Point Labels for the Finance, Commercial Services, and Technology Services Sectors 49
6.3 The 31 Key Momentum Drivers for the Finance Sector 52
6.4 Average Class Weights for ’max’, ’min’, ’both’, and ’none’ Categories Across Three Sectors. Weights < 1 give less importance during training. Weights > 1 give more importance during training. 53

6.5 Optimal Hyperparameters: Random Forest and Gradient Boosting Classifiers for Finance, Commercial Services and Technology Services Sectors . . . 54

6.6 Comparison of Random Forest (RF), left, and Gradient Boosting Machine (GBM), right, Classification Results for Finance, Commercial Services, and Technology Services Sectors. Precision, Recall, F1-score, and Support values are shown for each category (max, min, both, none) across both models. 55

A.1 The 20 Removed Non-Double-Type Columns and Their Definitions 65

A.2 The 309 Features Dropped from the Finance Sector data due to Missing Values Thresholding 74

A.3 Features Dropped from Finance Sector data due to Low Variance in Selected Stock Sample 75

A.4 Top Quintile of Features given Current Stock Sample from the Finance Sector Obtained Through Random Forest Feature Importance. 1mth forward returns target variable 76

A.5 Features Dropped from Correlation Analysis given Current Stock Sample from the Finance Sector 76

A.6 Features That Share No Mutual Information with 1 Month Forward Returns Dropped given Current Stock Sample from the Finance Sector 77

A.7 A Set of Key Momentum Drivers Found for the Finance Sector. The result of the comprehensive feature selection process. 78

Chapter 1

Introduction

This chapter offers a thorough overview of the thesis, starting with an explanation of the motivations behind our research goals. After outlining the research objectives, it then briefly summarises the experiments conducted and the scientific contributions of our study. The chapter wraps up by describing the structure of the thesis.

This thesis investigates the development of a classification model that predicts whether there will be a momentum turning point for a given stock for the coming month, and if so, categorises the turning point as a minimum or a maximum. We leverage crucial fundamental features specific to each sector to help capture momentum turning points and predict changes in momentum ahead of time.

We begin by adopting a robust feature engineering procedure. For the construction of impactful features, we detail a useful approach which allows us to both increase the granularity of the raw fundamentals series from quarterly to weekly and capture trends in the fundamentals relative to the entire sector. Following this, we explain our different approaches to labelling momentum turning points in the price data, including dividing the price history into fixed intervals while strategically disregarding certain minimums and maximums within these intervals. We then proceed by reducing the original feature space using a random forest feature importance model, followed by correlation analysis and mutual information analysis. Finally, given the features and their labels, we propose two different classification models to try to predict these turning points ahead of time. The proposed methodology aims to deliver a series of weekly turning point signals for each stock, which could aid an investor in constructing a sector-based equity momentum portfolio.

This research was undertaken in partnership with Banking Science Limited under the guidance and supervision of Dr. Michal Galas, who provided direction, validated the methodology, and affirmed the market relevance of the study's focus.

1.1 Research Motivation

Stock price movements are notoriously difficult to predict given several different factors that drive them, including, but not limited to, company performance, market sentiment and industry trends. This poses significant challenges for portfolio asset allocation. One strategy that has gained traction to address this challenge is momentum investing, which hinges on the hypothesis that stocks trending in a certain direction will continue to do so over a prolonged period of time. This approach, which focuses on identifying and leveraging patterns in price data to predict trends and reversals, has caught the attention of many investors. However, the efficacy of momentum investing based purely on price patterns comes into question. Over time, the market's ability to spot and exploit these patterns has diminished[3], making it increasingly difficult to capture turning points accurately.

In light of these limitations, corporate fundamentals present a more promising avenue for predicting changes in price momentum. Fundamental data, which includes metrics such as earnings, revenue growth, and other measures of financial health, can provide deeper insights into a company's potential for sustained performance[4]. These indicators are less susceptible to the noise and volatility that often obscure price patterns, offering a more stable basis for predicting momentum turning points.

This research aims to contribute to the literature by integrating company fundamentals into momentum analysis. Specifically, we seek to capture momentum turning points as early as possible by using ML techniques on fundamental features in time series. We propose that our methodology will enable us to more accurately sort stocks based on their short-term future momentum, hence allowing us to develop a more robust predictive framework for momentum investing. This approach is also motivated by the need for a computationally efficient classification algorithm capable of handling data issues, obeying boundary conditions, and being widely applicable across various large-scale problems in finance. If successful, our methodology could play a critical role in improving trading strategies whilst filling a significant gap in the existing research, which has largely overlooked the potential of fundamental data in predicting turning points in stock momentum.

This research also emphasises the importance of feature engineering and feature space reduction in managing the vast number of fundamental indicators relevant to our momentum analysis. We carry out feature engineering to transform our raw data into more meaningful metrics, which allows for more interpretability in our potential momentum drivers. We also detail a feature space reduction procedure, which not only allows for investors to identify the most relevant factors that contribute to stock returns, but can also substantially improve the memory and time complexity of real-time trading systems.

We also want to compare the different classical ML approaches in their the potential to achieve superior investment returns and manage risk more effectively. Applying classification models to predict turning points provides a relatively simple systematic approach in discerning the certain stock price dynamics. This allows investors to make informed decisions based on the signals constructed from a model which uses a well-accepted and reliable ML framework. By comparing the different approaches, we aim to build reliable models that not only predict turning points, but also adapt to changing market conditions, thus achieving a higher degree of accuracy in our predictions.

While fundamental data is widely used in stock analysis, its application in momentum analysis, particularly for identifying momentum turning points, is under-explored in the literature. Most existing studies either focus on using technical indicators for momentum prediction or they use fundamentals in a broader context without explicitly targeting momentum. This gap in the literature suggests a potential opportunity to enhance stock momentum analysis, in particular through systematically incorporating and optimising fundamental features when predicting changes in stock momentum.

1.2 Research Objectives

To develop a classification algorithm for stock momentum turning points that is of practical use for investors, we must accomplish the following objectives:

1. **Define and identify momentum turning points.**

Clearly state an operational definition of a series of momentum turning points to ensure consistency and precision when identifying these. Design a robust approach to accurately detect the significant changes in stock momentum within historical price data.

2. **Identify important sector-specific features.**

Use historical data to carry out feature engineering and feature space reduction. Use these features as the input for the prediction models.

3. **Develop an appropriate methodology for predicting turning points.**

Create a forward-looking model that anticipates stock momentum turning points. Consider several factors when building the pipeline, such as balancing computational efficiency with model accuracy and using the appropriate amount of data for the predictions at each point in time.

4. **Evaluate model performance and ensure model robustness.**

Test and validate the model using relevant performance indicators, ensuring model reliability under varying market conditions.

1.3 Research Experiments

This study introduces a novel approach to predicting stock momentum turning points, which is structured into three distinct experiments:

Exploratory Analysis and Momentum Driver Feature Engineering

The investigation begins by engineering features from the raw fundamentals, which can capture the change in the fundamentals for a particular company in comparison to other similar-sized companies in the same sector. This prepares for subsequent steps in the methodology where the features are used to predict momentum turning points ahead of time.

Finding Ideal Turning Point Labels

The definition of a momentum turning point is motivated, which must capture the significant change in the direction of a price trend, resulting in a new price trend in the opposite direction. Using this operational definition, an algorithm is developed to accurately label these momentum turning points in the daily stock price data.

Identifying Key Momentum Drivers and Predicting Turning Points

The dimensionality of the engineered feature space from experiment 1 is reduced, and the labels from experiment 2 are aggregated. The resulting critical features and labels are combined and passed through two different hyperparameter-tuned tree-based classification models. The models are then evaluated, using the relevant performance metrics, in their ability to predict momentum turning points. Further steps taken to ensure robustness of results are also discussed.

1.4 Scientific Contributions

This thesis significantly contributes to the domain of quantitative trading and company fundamental analysis by filling important voids in the existing research and offering original methods to examine the link between the financials of a company and its stock price movements. The primary academic contributions of this study are:

1. *Comprehensive Feature Engineering and Selection Framework*: The process of performing feature engineering on extensive datasets and identifying the most significant features is not only relevant to financial problems but is also applicable to various challenges in the STEM fields. This contribution demonstrates the generalisability and utility of these techniques across different disciplines.

2. *Broad Applicability of Turning Point Detection*: The main approach taken to identify turning points is designed to work across different scales, from small datasets to large, complex ones, ensuring wide-ranging applicability. The research extends to numerous academic domains where identifying anomalous movements of variables is critical.

3. *Algorithmic Insights*: The deployment of advanced algorithms in the current context offers valuable insights into how such methods can be effectively utilised to solve complex financial problems, thereby advancing the field of financial analytics and ML applications in finance.

1.5 Thesis Structure

The remainder of this thesis is structured into the following sections:

Chapter 2: Background and Literature Review. The next chapter offers a comprehensive overview of the background and existing research on company fundamentals analysis and stock momentum. It starts by tracing the evolution of factor investing, emphasising the importance of fundamentals as predictors of stock prices. Following this, the chapter explores various supervised machine learning techniques employed in this study, such as random forest and gradient boosting machine classification algorithms, and their application to factor investing. Additionally, it discusses the processes of hyperparameter tuning and model evaluation. The chapter concludes with a review of traditional momentum investing strategies and relevant related work, providing essential context for the study.

Chapter 3: Exploratory Analysis and Momentum Driver Feature Engineering. This chapter presents the first experiment aimed at constructing drivers of stock momentum. It begins with a comprehensive data preprocessing section, covering essential steps such as database filtering, adjusting price data for corporate actions, and rigorous data cleaning to ensure the dataset's reliability. The chapter then moves onto describing a robust feature engineering procedure. This foundational work sets the stage for deeper analysis and model development in subsequent chapters.

Chapter 4: Finding Ideal Turning Point Labels. This chapter introduces the second experiment, dedicated to generating accurate daily momentum turning point labels for the price data. It begins by precisely defining a momentum turning point and elaborates on the concept of a series of momentum turning points. Building on these definitions, the chapter then outlines algorithms designed to systematically identify all turning points within a historical price series. This thorough approach provides the necessary groundwork for identifying turning points in momentum, which is crucial for the study's final experiment.

Chapter 5: Identifying Key Momentum Drivers and Predicting Turning Points. This chapter details the third and final experiment, which integrates the outcomes of the previous experiments to predict momentum turning points. The initial focus is on reducing the original feature space to uncover key momentum drivers. These are identified using techniques such as random forest feature importance, correlation analysis, and mutual information analysis. The labels from the previous experiment are then aggregated. The important features and the corresponding new labels are then fed into classification models. The objective is to predict whether maximum or minimum turning points will occur within the next four weeks, with predictions being updated weekly. The chapter also covers model performance evaluation using classification metrics, along with the application of hyperparameter tuning and cross-validation to optimise accuracy. This experiment represents the culmination of the study, bringing together all the elements to provide actionable insights into market momentum.

Chapter 6: Results. This chapter presents the results of the experiments conducted in the study. It begins by identifying the most important features by sector, highlighting the key drivers of momentum in different market segments. It then proceeds to evaluate each phase of the methodology proposed to obtain the out-of-sample stock momentum turning point predictions.

Chapter 7: Conclusions and Future Work. This chapter summarises the key findings of this study and their implications. It discusses the strengths and limitations of the research, providing a balanced view of its contributions and areas for improvement. The practical applications of the findings to equities trading strategies are explored, illustrating how the insights can be used in real-world trading strategies. Finally, the chapter identifies possibilities for further research, suggesting directions for future studies to build upon the results and address the identified limitations.

Chapter 2

Background and Literature Review

This chapter lays the groundwork by highlighting the necessity of the research in this study and outlining the limitations of previous related studies. It opens with a discussion of the relevant topics, starting with an overview of the evolution of factor-based investing models and the use of fundamentals in stock price prediction. It then proceeds to describe the main statistical techniques and machine learning algorithms employed in the study, before discussing existing literature on momentum investing, thereby showcasing the uniqueness of this research.

2.1 Factor Investing and Fundamentals as Stock Price Predictors

This research concentrates on applying a factor-based methodology to model shifts in stock momentum. Factors, in the context of equities investing, are variables that help explain the returns and risks that are associated with different stocks or stock portfolios. They can be used to construct models that help investors understand market behaviour, manage risks, and optimise portfolios. Factors can be divided into distinct categories to facilitate the creation of a diversified investment portfolio. These categories include, but are not limited to, market factors, which primarily involve investing based on overall market risk; sentiment factors, which capture the mood and attitude of investors; and fundamental factors, derived from a company's financial statements. The latter will be the main focus of our study.

The Capital Asset Pricing Model (CAPM), is a foundational single factor investing model that aims to describe the relationship between systematic risk and expected return for assets, particularly stocks. The model, developed by William Sharpe in the 1960s[5], is used to estimate an investment's expected return based on its risk relative to the market. The CAPM is based on the idea that investors need to be compensated in two ways, namely time value of money and risk.

$$E(R_j) = R_f + \beta_j \left(E(R_m) - R_f\right)$$

is the CAPM where:

- $E(R_j)$ and $E(R_m)$ are the expected returns of investment j and the market respectively
- R_f is the risk-free rate, meaning that $E(R_m) - R_f$ is the market risk premium
- β_j is the beta of investment j, a measure of its volatility relative to the market

The CAPM is based on several assumptions: beta is the sole relevant measure of risk, investors hold diversified portfolios, markets are efficient, and borrowing and lending occur at the risk-free rate. The simplicity of the model therefore comes at the cost of being unable to capture the nuances of stock returns. The Fama-French Three-Factor Model (1993)[6] suggests an improvement to this by incorporating additional size and value fundamental factors, but there are still a multitude of other factors affecting stock returns that haven't been used in these models. Various other studies have been conducted following this to explore the relationship between fundamental factors and stock prices.

Kaizoji and Miyano (2018)[7] analysed share prices during the 2008 financial crisis and found significant deviations from company fundamentals. Using data from approximately eight thousand companies, they discovered that prices were overvalued before the crisis, undervalued during the crisis, and realigned post-crisis. This study clearly underscores the importance of fundamentals in understanding stock price fluctuations during major financial disruptions.

Using fundamentals from a different standpoint, Beck et al. (2017)[8] analysed company price-to-book-value (P/B) ratios in the prediction of stock price mean reversion, distinguishing value strategies from mean reversion strategies. They showed that while stock prices mean revert, company fundamentals do not, giving value strategies an advantage by leveraging clearer signals from fundamental data. Also adding to the relevant literature, Zhang et al. (2018)[9] compared the correlation between accounting data and stock prices in China and Thailand. Their study found that earnings per share and book value per share had stronger explanatory power in Thailand, suggesting that fundamentals can have varying impacts on stock price depending on the economic climate. Huang et al. (2019)[10] then introduced 'fundamental momentum' by analysing trends in firm fundamentals, finding that by combining fundamental and price momentum strategies one can achieve superior returns, supporting the integration of fundamental analysis with technical analysis for more robust investment strategies.

We now explore the existing literature which highlights that the dynamics of company fundamentals can be used to explain momentum in stock prices. Robert Novy-Marx's work

(2015)[4] demonstrates that earnings momentum, particularly earnings surprises, is a critical predictor of price momentum. By controlling for earnings surprises, the volatility of price momentum strategies can be reduced without diminishing high average returns. Furthermore, Ahmed et al. (2018)[11] reveals that price momentum tends to reverse when it is not supported by fundamentals, and that conversely, stocks where past price performance aligns with fundamentals exhibit more substantial future momentum. Also supporting the fact that financial statement analysis is a valuable tool for enhancing momentum strategies, the study "What Can Explain Momentum? Evidence from Decomposition" (2022)[12] identifies firm fundamentals as the most promising explanation for the momentum anomaly, surpassing other theories such as prospect theory, mental accounting[13], and the anchoring effect[14]. The study shows that firm characteristics and residual momentum can mitigate momentum crashes, highlighting the significant explanatory power of fundamentals.

Introducing ML into the mix, Huang et al. (2022)[15] motivates the use of supervised ML for stock prediction based on company fundamentals data in our study. Using 22 years of financial data it was found that, of the models that were tested, random forest feature selection performed the best in comparison to the benchmark statistical models, demonstrating the potential of ML in enhancing fundamental analysis for investment decisions.

2.2 Machine Learning and Other Statistical Analyses

Machine Learning (ML) refers to algorithms that allow computers to learn from data and make predictions or decisions without explicit programming. It involves training models on large datasets to analyse unseen data effectively. In factor investing, ML is crucial for processing and analysing extensive financial datasets with multiple factors, which can be challenging for traditional statistical methods. ML algorithms are categorised into supervised, unsupervised, semi-supervised, and reinforcement learning. This study focuses on supervised learning, specifically using regression to reduce the feature space and classification for our predictions. Regression predicts continuous values based on input features and labels, while classification sorts data into predefined categories using labelled data to guide decisions.

2.2.1 Models

We proceed by providing some background to the statistical analyses and modelling that will be carried out in this study's experiments. Some of the factor-based models mentioned earlier, including CAPM and the Fama French model, assume that there is a linear relationship between stock returns and the factors that affect it. However, real-world financial markets are often complex and likely exhibit non-linear relationships between variables. To capture these complexities more effectively, we must to explore non-linear modelling approaches. Tree-based models, such as random forests and gradient boosting machines, are particularly well-suited for this investigation as they can naturally handle non-linear interactions between variables, offering a better understanding of the factors driving stock returns, and hence stock price momentum.

Decision Trees, Random Forests and Gradient Boosting Machines

Decision trees work by recursively partitioning the data into subsets based on the values of input features, resulting in a tree-like structure of decisions. The primary goal of a decision tree is to predict the target variable by learning simple decision rules that are inferred from the data features.

The structure of a decision tree consists of a root node, which contains the entire dataset and initiates the first split, decision nodes, which represent further feature based splits, and the terminal node representing the final output.

The decision tree algorithm works as follows:

Step 1. Start with the entire dataset at the root node. Select the best feature to split the data based on either the lowest impurity in the resulting child nodes (for classification tasks) or the largest mean squared error (MSE) reduction (for regression tasks).

Step 2. For the chosen feature and threshold, create a decision node and partition the dataset into left and right branches based on the threshold.

Step 3. Recursively apply the same process to each subset (the left and right branches) until a stopping criterion is met.

Step 4. For prediction, traverse the tree from the root node to a leaf node by following the decisions at each node based on the input feature values.

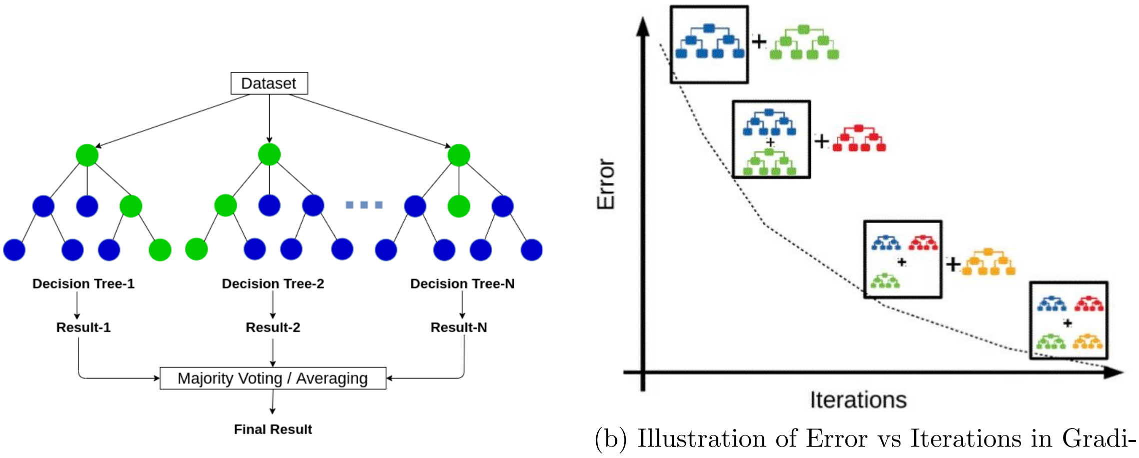
In classification tree models, the class prediction is typically determined by the majority class of the training samples that fall into that leaf node. However in regression tree models, the predicted value is typically the mean of the target values of the training samples that fall into that leaf node.

Note that the impurity metric we will be using for the feature splits for the classification model is the Gini impurity defined as:

$$Gini(p_j) = 1 - \sum_{j=1}^{n} (p_j)^2$$

where p_j is the probability of a data point belonging to class j, and n is the number of classes. For the regression model we split features based on the (MSE) metric as stated above.

Random forests enhance the performance of individual decision trees by creating an ensemble of trees and aggregating their predictions, though at the cost of greater required computational power and memory. During the training process the random forest algorithm creates multiple bootstrap samples (where a random subset of the data is sampled with replacement) from the original dataset. Each decision tree is then trained on one of these bootstrap samples, and the randomness of each sample helps to reduce overfitting, which is a common issue with individual decision tree models. The algorithm combines the results of multiple decision trees, using aggregation by majority voting for classification, or, for example, by taking an average for regression (see **Figure 2.1a**). This reduces variance of the model, making the final output more robust. Random forests are known for their ability to handle complex, non-linear relationships in the data. They also allow for the straightforward interpretation of feature importance.



(a) Illustration of a Random Forest Model[16]

(b) Illustration of Error vs Iterations in Gradient Boosting[17]

Figure 2.1: Visual Representations of the Random Forest Mechanism and the Iterative Error Reduction Process for GBMs

Unlike random forests, gradient boosting machines (GBMs) work by building 'weak learners', i.e. decision tree models, sequentially. GBMs create a series of decision trees, where each subsequent tree is trained to predict the residuals of the combined ensemble of all previous trees. The aim of a GBM is to minimise the prediction error at each step through the gradient descent algorithm, thereby iteratively improving the overall model accuracy (see **Figure 2.1b**).

The GBM multi-class classification algorithm works as follows:

Step 1. Starting with an initial ensemble model, calculate the log-odds of each class k for multi-class classification:

$$I_0^k(x) = \log\left(\frac{N_k}{N}\right)$$

where I represents the model's prediction function at a given stage in the boosting process, the subscript of I is the iteration number, the superscript is the class index, N_k is the number of samples in class k, and N is the total number of samples.

Step 2. For the total number of boosting iterations (from $i = 1$ to M): compute pseudo-residuals for each class k:

$$r_{im}^{k} = -\frac{\partial L(y_{ik}, I_{m-1}^{k}(x_i))}{\partial I_{m-1}^{k}(x_i)} = y_{ik} - p_{m-1}^{k}(x_i)$$

where y_{ik} is 1 if y_i is in class k and 0 otherwise, and $p_{m-1}^{k}(x_i)$ is the predicted probability for class k.

Step 3. Fit a decision tree $h_m^k(x)$ to the residuals.

Step 4. Update the model by adding the new tree, scaled by a learning rate η:

$$I_m^k(x) = I_{m-1}^k(x) + \eta h_m^k(x)$$

Step 5. Output the final prediction. Convert model outputs into probabilities, e.g., using the softmax function, and output the predicted class with the highest probability:

$$\hat{p}^k(x) = \frac{e^{I_M^k(x)}}{\sum_{j=1}^{K} e^{I_M^j(x)}}; \quad \hat{y}_i = \text{argmax}_k \hat{p}^k(x)$$

where K is the total number of classes.

GBMs offer several advantages over other models such as random forests and individual decision trees due to their iterative sequential model training. However this comes at the cost of longer training times and more susceptibility to overfitting. It is therefore crucial to consider the balance between time-efficiency and performance when applying GBM models, and to more carefully tune hyperparameters and validate GBMs to ensure performance reliability.

Mutual Information Regression

Mutual information (MI) quantifies how much information about the target variable is gained by knowing the feature, thus helping in identifying the most informative features for predicting the target variable. Mathematically the mutual information between two variables X and Y can be calculated as

$$I(\hat{X}; \hat{Y}) = \sum_{x \in \hat{X}} \sum_{y \in \hat{Y}} P(x, y) \text{log}\left(\frac{P(x, y)}{P(x)P(y)}\right)$$

where P(x) and P(y) are the marginal probability distributions of the features and the target variable respectively, and P(x, y) is the joint probability distribution of the features with the target variable (note here we define $\hat{X}$ and $\hat{Y}$ to be the sets of instances of X and Y respectively). The MI regression will return a score for each feature based on the average mutual information across all samples, indicating its dependency on the target variable, where higher MI scores suggest a stronger relationship between the feature and the target.

2.2.2 Hyperparameter Tuning

Hyperparameters are the parameters of a model that are defined before the learning process begins. Some examples include the number of trees in a random forest, or the learning rate in gradient boosting. Proper tuning of hyperparameters can significantly enhance a model's performance by finding the optimal settings that minimise errors and improve overall model accuracy. There could also be several hyperparameters in a single model that require optimisation. For this, different hyperparameter tuning methods exist, such as Random Search, Bayesian optimisation and Grid Search. In this investigation, the focus is on the latter.

After identifying the hyperparameters to tune and specifying the range or distribution for each hyperparameter, the Grid Search method involves systematically selecting and evaluating every possible combination of hyperparameter values from the predefined grid. The optimisation algorithm then continues as follows (as it does for all other tuning methods):

Step 1. For each set of hyperparameters, train the machine learning model using the training dataset.

Step 2. Evaluate the model's performance on the validation set using an appropriate metric.

Step 3. Continue sampling hyperparameters and evaluating the model for a predetermined number of iterations or until computational resources are exhausted, keeping track of the hyperparameter configuration that yields the best performance.

Step 4. After completing the search, select the hyperparameters that achieved the best performance on the validation set.

We choose the Grid Search tuning method for this investigation because of its thoroughness in exploring all possible combinations of hyperparameters, ensuring that we identify the optimal configuration. This exhaustive approach is particularly valuable for our study, as it provides comprehensive coverage of the hyperparameter space, which is essential given the complexity of the models we will be using.

2.2.3 Model Evaluation

Performing rigorous model evaluation involves being able to accurately assess the performance of a trained model, using various metrics and techniques, to determine how well it generalises to unseen data. Ensuring the robustness of the obtained results is vital to confirm that the model is reliable and applicable in real-world scenarios. Several measures are taken in this investigation order to achieve this.

Validation and Cross-Validation

Generalisation to unseen data is crucial to ensure the applicability of models in real-world scenarios. This involves splitting the dataset into distinct sets: training, validation, and test sets. The training set is used to fit the model, the validation set is used to tune hyperparameters and evaluate model performance during the development phase, and the test set, which remains unseen until the final evaluation, provides an unbiased estimate of the model's effectiveness. Without proper validation, models can achieve high accuracy on training data but fail to perform well on new, unseen data, leading to misleading conclusions about their predictive power.

Cross-validation is a robust model evaluation technique that provides a more reliable assessment than a single train-validation-test split. In k-fold cross-validation, the dataset is divided into k equally-sized folds. The model is trained and evaluated k times, each time using a different fold as the validation set and the remaining folds for training. This reduces variance of the model output and ensures consistent model performance across different data subsets. For time series data however, where temporal dependencies must be respected, rolling-forward cross-validation must be employed. This method involves iteratively training the model on an expanding window of past data and validating it on a fixed window of future data, effectively mimicking real-world scenarios where future data is not available at training time. As each iteration progresses, more historical data is included in the training set, which helps in capturing trends and patterns more effectively.

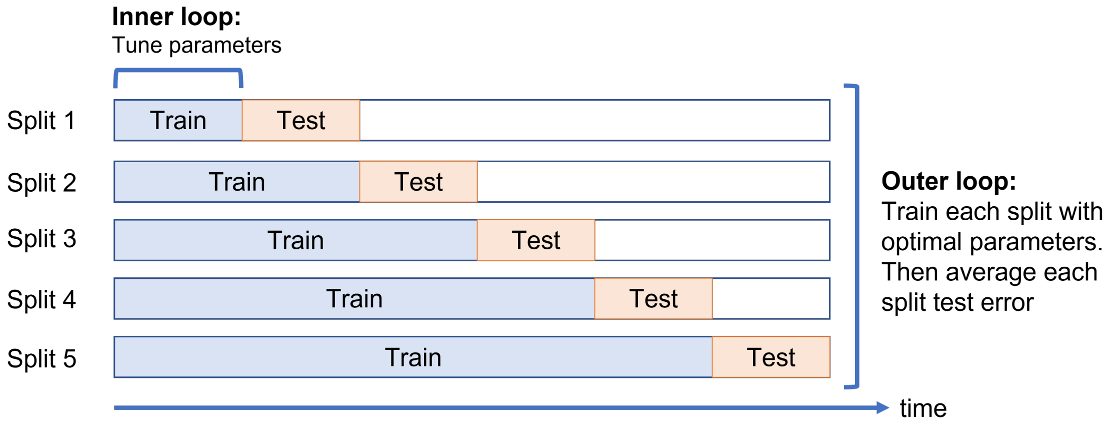


Figure 2.2: Illustration of Rolling-Forward Cross-Validation with Expanding Training Folds[2]

Model performance is evaluated by, for example, taking the average of the performance metric across all the testing folds. Although this approach is more computationally intensive than the single split method, it results in better estimates of model performance due to lower variance in the relevant metric.

Defining Relevant Metrics

The Root Mean Squared Error (RMSE) is a widely used error metric for evaluating the performance of regression models. It measures the average magnitude of the errors between the predicted values and the actual values in a dataset, calculated through the formula:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

where n is the number of observations, y_i is the actual value for the i-th observation and $\hat{y}_i$ is the predicted value for the i-th observation. RMSE will be used as the performance metric for the regression model in this investigation given its natural focus on minimising large errors, and given that it can be expressed in the same units as the target variable, making it easy to interpret.

In multi-class classification problems, precision, recall, and F1 scores are important metrics for evaluating model performance. They provide insights into how well a model can distinguish between multiple classes. Denote FP := ”False Positives”, TP := ”True Positives”, FN := ”False Negatives”, TN := ”True Negatives”.

Class-Specific Definitions	**Weighted Definitions**
PRECISION $\text{Precision}_k = \frac{\text{TP}_k}{\text{TP}_k + \text{FP}_k}$	WEIGHTED PRECISION $\text{Weighted Precision} = \frac{\sum_{k=1}^{K} n_k \times \text{Precision}_k}{\sum_{k=1}^{K} n_k}$
RECALL $\text{Recall}_k = \frac{\text{TP}_k}{\text{TP}_k + \text{FN}_k}$	WEIGHTED RECALL $\text{Weighted Recall} = \frac{\sum_{k=1}^{K} n_k \times \text{Recall}_k}{\sum_{k=1}^{K} n_k}$
F1 SCORE $\text{F1 Score}_k = \frac{2 \times \text{Precision}_k \times \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k}$	WEIGHTED F1 SCORE $\text{Weighted F1 Score} = \frac{\sum_{k=1}^{K} n_k \times \text{F1 Score}_k}{\sum_{k=1}^{K} n_k}$

Table 2.1: Comparison of Class-Specific and Weighted Metrics

where n_k is the number of true instances for class k and K is the total number of classes. The weighted metrics are particularly important because they account for the class distribution in the dataset, ensuring that the performance of the classification algorithm is fairly evaluated across all classes. We avoid bias towards majority classes and ensure that the evaluation reflects the model's effectiveness in handling both frequent and rare classes.

Precision indicates how many of the predicted positive cases for a particular class are actually correct. A high precision score means that most of the instances predicted as positive are indeed positive. It is particularly important in applications where false positives are costly or undesirable. Meanwhile, recall measures the model's ability to identify all relevant instances of a class. A high recall indicates that most actual positive instances are captured by the model. It is critical in scenarios where missing a positive case is more detrimental than producing false positive results. The F1 score combines precision and recall into a single metric using the harmonic mean, which is useful when both FP and FN carry significant costs.

Tackling Class Imbalance

When constructing classification models, it is very important to manage class imbalance when it exists. Class imbalance refers to a situation in classification problems where the number of instances in one class is significantly higher or lower than in other classes. This imbalance can lead to biased model predictions for minority classes, as the model might be influenced by the more frequent classes.

Adjusting class weights is an effective solution for tackling class imbalance in machine learning models. By assigning higher weights to minority classes and lower weights to majority classes, the model is encouraged to pay more attention to underrepresented classes during training. This helps to mitigate bias, leading to improved performance across all classes. Additionally, it avoids the potential pitfalls of oversampling or undersampling, such as data duplication or loss of information, preserving the dataset's integrity while still addressing the imbalance.

2.3 Traditional Momentum Investing and Related Work

Momentum investing is based on the hypothesis that stock prices moving in one direction will keep on moving in that direction for a certain period of time. In the existing literature there seems to be a lack of consensus on how momentum turning points can be effectively identified using the stock price history. The literature also does not provide a clear approach for pinpointing momentum drivers that are specific to individual sectors.

2.3.1 Momentum Metrics and the Momentum Factor

We must first note that the technical indicators of momentum such as the relative strength index (RSI) and the moving average convergence-divergence (MACD) are typically used for shorter-term analysis in identifying overbought or oversold conditions in the market, which might not always lead to substantial trend reversals. In light of this, researchers have tried to come up with more reliable longer term momentum signals, which when combined, are likely to be more profitable in the long term as well.

Dual momentum investing, introduced by Gary Antonacci in 2014, is a strategy that combines two types of momentum: relative momentum (cross-sectional momentum) and absolute momentum (time-series momentum). In the context of equities investing, relative momentum compares the performance of different stocks to each other over a given period, where stocks with the highest relative performance are selected for potential investment. On the other hand, absolute momentum evaluates the performance of the stock against its own historical performance by considering, for example, whether the stock's returns are positive over a given period, and if so these may be considered for investment. Here are examples of these absolute and relative measures:

12-1 Month Momentum:

$$\text{12-1 month momentum} = \text{cum. return over past year} - \text{cum. return over past month}$$

Absolute:
For a single stock, if the above momentum is positive, it is a signal to **buy**. If the momentum is negative, it is a signal to **sell**.

Relative:
When calculating the 12-1 month momentum for several different stocks, we interpret the results in relative terms:

- A signal to **buy** is generated if a particular stock lies in the **top decile** of stock momenta (indicating strong positive momentum).
- Conversely, a signal to **sell** is generated if the stock lies in the **bottom decile** (indicating strong negative momentum).

Note that the return for the most recent month is excluded to account for potential short-term reversal effects.

Antonacci highlighted in his 2016 paper[18] that while both absolute and relative momentum can enhance returns, absolute momentum plays a more significant role in reducing volatility and drawdowns. He also mentions how a combination of absolute and relative momentum provides the optimal balance, delivering superior returns while managing risk effectively. Equity investors can therefore, for example, construct a portfolio of stocks that meet both momentum criteria.

Pedro Barroso's research (2015)[19] addresses the variability and predictability of momentum risk over time. He highlights that traditional momentum strategies, despite their high Sharpe ratios, suffer from severe crashes. By managing momentum risk, Barroso demonstrates a significant improvement in performance and a reduction in the adverse effects of momentum crashes, which is crucial for making momentum strategies more appealing to risk-averse investors. Jusselin et al. (2017)[20] also explore the momentum risk premium, highlighting the hedging properties and diversification benefits of momentum strategies. This theoretical foundation is supported by Thierry Roncalli (2017)[21], who emphasises the importance of diversification asymmetry in momentum investing, particularly during economic downturns, aligning with Antonacci's dual momentum and risk management concepts. In a complementary approach, He et al. (2017)[22] explore time series momentum and reversal strategies in "Asset Allocation with Time Series Momentum and Reversal." They develop a continuous-time asset price model that combines market fundamentals with timing opportunities based on market trends and volatility. The empirical results show that integrating short-term momentum with reversal strategies significantly outperforms traditional strategies, highlighting the need to consider both momentum and reversal in strategy formulation. Expanding on the risk management theme, Li et al. (2022)[23] investigate the downside risks of momentum strategies in "Momentum and the Cross-Section of Stock Volatility". They attribute these risks to the cross-sectional volatility of individual stocks. Their proposed Generalised Risk-Adjusted Momentum (GRJMOM) method adjusts for high momentum-specific risks, proving to be more profitable without increased risks. This study therefore emphasises the importance of managing cross-sectional volatility to enhance momentum strategy performance.

2.3.2 Momentum Turning Points

The concept of momentum turning points is critical to the efficacy of time-series (TS) momentum strategies, as highlighted by Goulding et al. in their 2023 paper[1]. They address the inherent challenge of time-series momentum strategies, balancing the responsiveness to trend changes with the noise in signals. Traditional slow momentum signals fail to react promptly to trend reversals, whilst fast signals often misinterpret noise as genuine trends, resulting in false alarms. Their research proposes an optimal dynamic speed selection strategy that combines slow and fast momentum strategies to capture these turning points more effectively. The findings reveal that this strategy achieves better out-of-sample performance by integrating both slow and fast signals.

We take a different approach by directly identifying and predicting these critical turning points without relying on traditional momentum signals, leveraging feature engineering, selection and classification algorithms to detect momentum turning points in real-time.

Chapter 3

Experiment 1 - Exploratory Analysis and Momentum Driver Feature Engineering

This chapter focuses on constructing sector-relative features that potentially drive stock momentum within each market sector. It details the critical steps involved in data acquisition and preprocessing, and experiments with different feature engineering procedures. The chapter concludes with a summary that lays the groundwork for subsequent analyses.

We use data provided by FactSet through Banking Science Limited for the analysis in this study. The original data was compiled on 18th June 2024. We acquire the relevant data from the FactSet database using PySpark SQL and write all of our code within Jupyter notebooks in a JupyterHub setup connected to a Hadoop Distributed File System (HDFS). Where possible, we distribute the data and computational workload across the twenty nodes of Banking Science's computer cluster.

3.1 Data Preprocessing

We carry out a rigorous preprocessing of our financial data, divided into the filtration and cleaning of Historical Daily Prices and Quarterly Earnings Reports, and then combined preprocessing for alignment and consistency. To obtain data relating to daily prices, we source from tables containing information prices, dividends, splits, and many other price-related factors, adjusting prices for corporate actions and correcting price anomalies. To obtain earnings reports data we filter the database to exclude companies in irrelevant sectors or companies that have inconsistent financial histories with the price data. We address missing and anomalous values similarly to the price data, remove outdated values and standardise dates in both dataframes to prepare for feature engineering.

3.1.1 Database Filtering

We must first consider the portion of extensive FactSet database that is relevant for our analysis. FactSet provides detailed company information, including Historical Daily Prices (HDP) and Quarterly Earnings Reports (QER), on a wide array of companies globally, that trade on several different exchanges. Given the extensive amount of data available for companies traded in North America in the ’US dollars’ currency denomination, we choose to focus our analysis on this specific subset of the data. More specifically, we filter for stock data corresponding to the period from 1st January 2000 to 18th June 2024, analysing companies with a market capitalisation of over 500 million US dollars listed on the New York Stock Exchange (NYSE) or NASDAQ. Data acquisition of stocks from the year 2000 onwards generally results in more completeness and reliability given advancements in data collection and reporting standards. Also, focusing on relatively larger companies helps in reducing the noise in the data that may arise from the erratic performance of smaller, less established companies. In total we end up with close to 6750 stocks for this study, each with roughly 97 observations per fundamental.

We choose to analyse the companies using a sector-based approach. Companies within the same sector tend to share common characteristics, regulatory environments and market conditions. This division by sector allows for a more granular and meaningful evaluation of companies by accounting for their shared attributes, thereby enhancing the robustness and relevance of our research findings. FactSet provides data on 21 different sectors, but for the purpose of this analysis, we exclude the ’Government’ and ’Miscellaneous’ sectors.

Stock Market Sectors		
Finance	Health Services	Process Industries
Commercial Services	Health Technology	Producer Manufacturing
Communications	Industrial Services	Retail Trade
Consumer Durables	Non-Energy Minerals	Technology Services
Consumer Non-Durables	Energy Minerals	Transportation
Consumer Services	Distribution Services	Utilities
Electronic Technology		

Table 3.1: The 19 Significant Stock Market Sectors

The ’Government’ sector is excluded because it primarily consists of entities that are subject to unique regulatory frameworks and operational constraints, which differ significantly from those in the private sector. These differences would introduce inconsistencies into our analysis, making it difficult to draw meaningful comparisons with other sectors. Companies in the 'Miscellaneous' sector are very diverse since they include companies that do not fit neatly into other defined sectors. This would complicate the development of our meaningful and coherent analysis. Hence, we focus our study on the 19 remaining sectors, with the intent of carrying out the relevant data analysis one sector at a time.

We use two primary categories of database tables provided by FactSet: Historical Daily Prices (HDP) and Quarterly Earnings Reports (QER). These naturally split the data preparation process into two distinct sections: prices and company fundamentals, the datasets for which will be combined later for subsequent analysis. The HDP tables provide daily price information on the securities being traded. They ensure that only actively covered securities are analysed, and they provide information on dividend payouts and stock splits for accurately calculating the adjusted closing price series for each stock.

Historical Daily Prices	**Quarterly Earnings Reports**
fp_basic_prices	ff_basic_qf
fp_basic_dividends	ff_basic_der_qf
fp_basic_splits	ff_advanced_qf
ff_basic_qf	ff_advanced_der_qf
sym_entity_sector	sym_entity_sector
fp_sec_entity	fp_sec_entity
sym_coverage	sym_coverage
factset_sector_map	factset_sector_map

Figure 3.1: Historical Daily Prices tables and Quarterly Earnings Reports tables to be queried

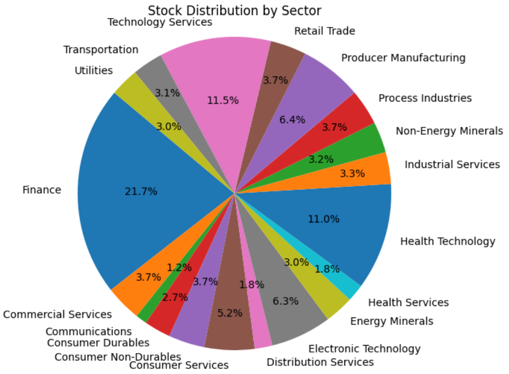


Figure 3.2: Pie Chart of Stocks by Sector

The QER tables focus on the core financial metrics, such as earnings and revenue, corresponding to each security. They also provide deeper insights through derived fundamentals, allowing for a more nuanced analysis of the financial health of different companies, and hence their resulting stock price dynamics. The dataset obtained from the QER tables contains 600 different company fundamental metrics organised in time series for each stock.

We ensure that the unique stock identifiers corresponding to both HDP and QER are the same. The distribution of proportion of stocks in each sector as a percentage of the total is shown in the pie chart in **Figure 3.2**. We illustrate the data cleaning, manipulation, and analysis process for the Finance sector (the overall procedure is the same for all other sectors), having sourced the data from the various tables mentioned previously in **Figure 3.1**.

3.1.2 Adjusting the Price Data for Corporate Actions

Adjustments ensure that the HDP data is consistent and comparable over time. Corporate actions such as stock splits, dividends, and spinoffs can significantly alter the nominal price of a stock. By adjusting for these events, we can maintain consistent price series that reflect the true values of the stocks at each point in time. Having retrieved the data for daily prices, dividends and splits for each company, we gather and organise the data to facilitate the calculation of various adjustment factors, which are essential for obtaining accurate, adjusted prices over time.

The concept of stock splits plays a crucial role in this adjustment process. An m-for-n stock split converts x shares at price y into $\frac{m}{n} \times x$ shares, each at a price of $\$(\frac{n}{m} \times y)$. Note that each share may produce more shares e.g. every 1 share becomes 5, but the converse is also possible - after a stock split, 5 existing shares could also become 1. Companies may systematically issue stock splits if the price increases too much in order to make the purchase of the company's shares more manageable and to introduce liquidity.

To account for splits, a cumulative split factor is calculated. We create temporary columns to store the intermediate values for the calculation. Working backwards from a cumulative split factor of 1 for the most recent data point, we carry this value backwards until the day of the previous split.

Day	**Desc.**	**Raw SF**	**Cumulative SF**	**Unadj. → Adj. # Shares**	**Unadj. → Adj. Price ($)**
d_0	No split	1	1	1 → 1	10 → 10
d_{-1}	4-for-1	0.25	1	4 → 1	10 → 10
d_{-2}	No split	1	0.25	4 → 1	40 → 10
d_{-3}	No split	1	0.25	4 → 1	40 → 10
d_{-4}	3-for-2	0.667	0.25	6 → 1	60 → 10
d_{-5}	No split	1	0.167	6 → 1	60 → 10

Table 3.2: Example price adjustment over time using split factors (SFs)

We multiply this value by the split factor on this date to obtain the previous cumulative split factor, and we iteratively carry out this process until we reach the beginning of the stock history (see **Table 3.2**). The split-adjusted price can then be calculated on each date as

$$\text{split-adjusted price} = \text{cumulative split factor} \times \text{unadjusted price}$$

Dividend adjustments are also important: the dividend split factor is applied to adjust the price for any dividends issued on the ex-dividend date, ensuring that the price reflects the true value of the stock after accounting for any dividend payouts. A split-adjusted dividend can be calculated similarly by multiplying the unadjusted dividend with the cumulative split factor. Note that FactSet provides both the split factors and the dates on which the splits occur through the table *fp_basic_splits*.

We must also account for spinoffs when adjusting prices. This is where a company distributes shares of a subsidiary to its shareholders, which in turn affects the stock price of the parent company. We can calculate the spinoff factor through the formula

$$\text{spinoff factor} = \frac{\text{unadjusted price} - \text{special dividend}}{\text{unadjusted price}}$$

and the cumulative spinoff factor can be calculated through the same iterative procedure as the cumulative split factor, but using spinoffs in place of splits. The spinoff-adjusted price can be calculated on each date as

$$\text{spinoff-adjusted price} = \text{cumulative spinoff factor} \times \text{unadjusted price}$$

and we can obtain an entire spinoff-adjusted price history for each stock. Note that FactSet provides information on the amount of the dividend issued on the ex-date, the dividend class (whether it is a regular dividend or a special dividend) and the dividend amount through the table *fp_basic_dividends*.

We combine the aforementioned formulas together to calculate the fully-adjusted price series. Note that to reduce the number of computations required for this, we calculate the split-adjusted price first (using the expression in the brackets), and then multiply this by the cumulative spinoff factor to obtain the fully-adjusted price:

$$\text{fully-adjusted price} = \text{cum. spinoff factor} \times (\text{cum. split factor} \times \text{unadjusted price})$$

We now have a price series that fully accounts for corporate actions and is a true reflection of the actual share prices of the companies at the close on any given day.

3.1.3 Data Cleaning

Aligning the Fundamental and Price Datasets

The next step in our preprocessing pipeline is to identify and remove companies or periods where the data is incomplete or misaligned between the fundamentals and prices, thus improving the overall quality of our datasets.

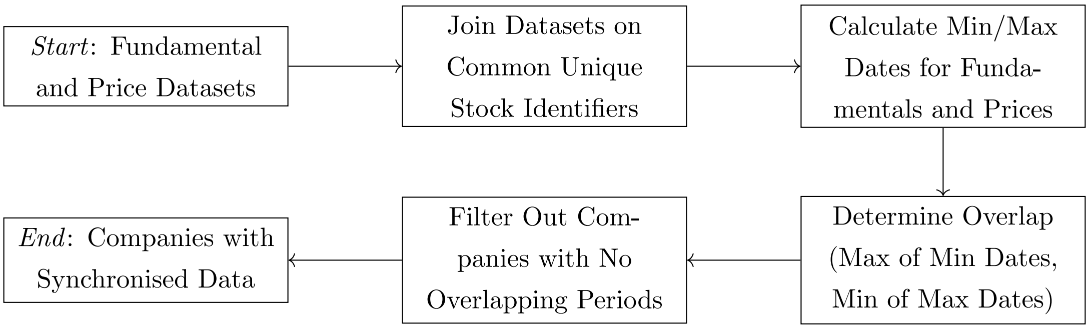

Start: Fundamental and Price Datasets
Join Datasets on Common Unique Stock Identifiers
Calculate Min/Max Dates for Fundamentals and Prices
Determine Overlap (Max of Min Dates, Min of Max Dates)
Filter Out Companies with No Overlapping Periods
End: Companies with Synchronised Data

Figure 3.3: Pipeline for Aligning Fundamental and Price Data

We begin by temporarily joining the fundamental and price datasets on the common unique stock identifiers, ensuring that the datasets can be combined correctly for each company. We calculate the minimum and maximum dates available in both the fundamental and price datasets. This determines the periods during which each type of data is available for a particular company. Next we find the maximum of the minimum dates and the minimum of the maximum dates for each company. We use this to identify the exact overlap period during which both fundamental and price data are available, ensuring that the datasets are aligned temporally. We filter out companies with no overlapping periods between their fundamental and price data, ensuring that only companies with synchronised data are retained for analysis, improving the integrity of the dataset.

Replacing Anomalies in the Price Data

Price data for stocks can be affected by various anomalies and noise due to market volatility, data entry errors, or irregular trading activities. These anomalies can significantly skew analysis results if not addressed. We therefore employ a function to correct the price data by identifying and replacing these anomalies.

Day	Adj. Price ($)	Moving Avg. ($)	Abs. Deviation ($)	Anomaly?	Corrected Price ($)
1	100	100	0	No	100
2	101	100	1	No	101
3	102	100	2	No	102
4	150	101	49	Yes	102.5
5	103	102	1	No	103
6	102	103	1	No	102
7	103	103	0	No	103

Table 3.3: Illustration of Price Anomaly Detection and Correction for Single Stock

We start by calculating the moving average and moving standard deviation for each stock price series over a 10 day rolling window, which helps in identifying the expected range of price values based on recent data. We then compute the absolute deviation of each price point from the moving average. If the deviation exceeds four times the moving standard deviation, we flag the price point as an anomaly. Using this thresholding, we count the number of anomalies to be 4603 across the 2.64 million price points for stocks in the Finance sector, meaning on average around 0.17% of price data points are classed as anomalies. For each identified anomaly, we replace the anomalous price with the average of the previous and next price values. We interpolate to ensure that the corrected price value is consistent with the surrounding data points.

We now have a corrected, corporate action-adjusted closing price series for each stock.

Using Latest Revisions and Dealing with Missing Values in the Fundamentals

In order to use data of the highest quality for our models, we should ensure that we are using the latest revision of the fundamentals provided by FactSet, and we must come up with a way to systematically handle the large number of missing values that may be present.

Using the ff_update_type column, we filter out any outdated data, ensuring that only the most recent and accurate data is retained, reflecting the latest revisions. We now proceed to deal with missing values by looking at the fundamentals dataframe both row-wise and column-wise. We begin by dropping any rows containing all missing values. If even one of the columns is non-empty on that particular date, it can be of use for our model. We then eliminate any dataframe columns that have more than 30% missing values. Prior to this step, the dataset had 601 features, with an average proportion of missing values per column at approximately 43.7% for the Finance sector. After this filtration, the number of features drops significantly to 276, and the average proportion of missing values per column decreases to around 7.14%, further enhancing the dataset's quality. These statistics are similar for the other sectors, where we end up with around 4-10% missing values on average per feature per sector. We address the remaining missing values after applying the necessary transformations for feature construction.

3.2 Feature Engineering

3.2.1 Approach 1: The Rate-of-Change Transformation

The conversion of fundamental data into quarter-over-quarter (QoQ) and year-over-year (YoY) rate-of-change (ROC) could be a pivotal step in transforming our dataset for further analysis. This may allow us to capture the longer-term dynamic changes in the fundamental features over time, which may be crucial in understanding the growth and performance trends of companies. It also helps in making the data more suitable for time-series analysis and predictive modelling, as it highlights the velocity and acceleration of changes in the fundamental metrics. These rates of change are defined as:

$$\text{QoQ ROC} = \frac{F_t - F_{t-1}}{F_{t-1}}, \qquad \text{YoY ROC} = \frac{F_t - F_{t-4}}{F_{t-4}}$$

The ROC transformation effectively doubles the number of columns in the dataset, creating a more comprehensive feature set from the original, resulting in roughly 550 features. However, this can potentially generate infinite values, particularly in cases where the denominator in the rate calculation is zero. We may have to use interpolation to replace the infinite values in the training folds and forward fill the infinities with the previous finite value (to avoid lookahead bias) in the testing folds. Luckily in our case, no infinite values were found for any of the market sectors. We fill the remaining missing values in our rate-of-change dataframe with zeros, assuming no growth in the fundamentals over the given period, which is our best guess given that these values are unknown.

Given, however, that we want weekly turning point predictions for the next month, we realise that quarterly features are not really of much use since they cannot capture several turning points occurring within the same quarter. In this study we therefore use an alternative approach which allows us to construct features of a weekly granularity.

3.2.2 Approach 2: Sector-Relative Feature Engineering

Here we outline the alternative, likely better, approach to constructing features, that is engineering sector-relative metrics from fundamental data to capture the relative performance of stocks within their respective market value categories. This process not only allows us to account for differences in company size, but also enhances the sensitivity of the fundamental data by aligning it with a more frequent, weekly timeframe rather than the typical quarterly updates.

Company (fsym)	ff_mkt_val (USD)	mktval_category	ff_eps_rpt_date	Aligned Date (Friday)	Metric (e.g., Revenue)	Sector Avg. by mktval_category	Relative Feature (Difference)
A	15 billion	Large (3)	2023-01-03	2023-01-06	1.5 billion	1.4 billion	0.1 billion
B	25 billion	Large (3)	2023-01-05	2023-01-06	2.0 billion	1.4 billion	0.6 billion
C	9 billion	Medium (2)	2023-01-02	2023-01-06	1.2 billion	1.1 billion	0.1 billion
D	8 billion	Medium (2)	2023-01-04	2023-01-06	1.0 billion	1.1 billion	-0.1 billion

Table 3.4: Illustration of Sector-Relative Feature Engineering for the Revenue Feature

We assign each set of fundamentals to its actual earnings release dates by replacing each original fiscal period end date with the respective report date, ff_eps_rpt_date, ensuring that the fundamentals reflect the date when investors and analysts first have access to the updated financial information.

To contextualise each company's financial metrics, we introduce a market value categorisation. This involves creating a new column, mktval_category, which classifies each company based on its market value (ff_mkt_val) into three distinct categories: small, medium, and large. Companies with a market value below a defined threshold (e.g. 2 billion USD) are categorised as small, those above a higher threshold (e.g. 10 billion USD) are categorised as large, and the rest are classified as medium.

Now we note that earnings reports for different companies are released on various dates, which could cause inconsistencies when comparing data across companies or when trying to aggregate data over time. To maintain consistency, we use the earnings reports released every Friday, ensuring that the financial data is aligned on a weekly basis. We compute weekly averages of financial metrics across companies in the same market value category, creating a sector-relative benchmark. For each company, we compare its most recent earnings report (aligned to the nearest Friday) to these weekly averages. The differences between the static quarterly metrics and the dynamic weekly averages are stored as new values. Only features constructed through this approach will be used for subsequent models in this study.

We fill the remaining missing values in our relative features dataframe with zeros, assuming no difference compared to the sector-average benchmark, which is, again, our best guess given that these values are unknown.

Feature construction is demonstrated in **Table 3.4**. The table is divided into two parts. The first part contains the columns for the company symbol, market value, market value category, earnings report date, and aligned date. The second part contains the columns for the financial metric, sector-relative weekly average, and the engineered feature. We end up with roughly 1100 observations per feature per stock.

3.3 Summary

This chapter establishes the foundation for this study's investigation by detailing a robust and well-structured procedure to obtain a high-quality dataset and construct potential drivers of stock price momentum. The comprehensive preprocessing pipeline will ensure the data's integrity and usability, setting the stage for deriving meaningful insights and producing well-informed results. The chapter adopts a systematic approach, progressing from data preprocessing to feature engineering, ultimately producing a refined dataset for in-depth analysis. Key elements covered include:

1. *Data Preprocessing*: The chapter begins by detailing the filtration of the FactSet database. The focus of the analysis is on stocks listed on the NYSE and NASDAQ with a market capitalisation of over 500 million USD, spanning from January 2000 to June 2024. A sector-by-sector analysis is conducted. Rigorous cleansing is applied to ensure data alignment and consistency, including adjustments for corporate actions and the identification and correction of price anomalies. Missing values are systematically addressed, and outdated data is removed to maintain the highest data quality. The result is a clean and consistent dataset that forms the basis for subsequent feature selection and model training in later chapters. Note that there were no duplicate rows or any infinite values in the feature columns.

Action on Finance Sector Data	**Before**	**After**
Prices data: replacing anomalous values	4603 (~0.17% of all prices)	0 – replaced anomalies with avg of [prev, next]
Fundamental data: Drop columns with missing value proportion >30%	Avg proportion of nan values per column: ~43.7%	~7.14% (filled with 0s before removal of low var features assuming no difference to sector avg, giving 0% nans)

Table 3.5: Cleansing Stats for Finance Sector

2. *Feature Engineering*: The chapter explores two approaches to feature engineering.

The first approach involves rate-of-change transformations to capture dynamic changes in fundamentals, while the second, more refined, approach focuses on sector-relative feature engineering. By aligning fundamental data with weekly earnings dates and computing differences to sector-relative benchmarks, this method enhances the sensitivity of the features, providing a more granular view for further analysis.

The outcome of this chapter is the carefully curated prices and features datasets, consisting of around 6750 stocks across all sectors. The prices datasets are made up of millions of corporate action-adjusted closing prices, and the features dataset is composed of 276 engineered features with, on average, roughly 1150 observations per feature. These datasets are now primed for subsequent modelling and analysis.

Chapter 4

Experiment 2 - Finding Ideal Turning Point Labels

This chapter begins by succinctly defining the concept of momentum and momentum turning points. It then proceeds to find an ideal way to label momentum turning points in the stock price data by testing two two promising algorithms. The key points are then summarised, setting the stage for the modelling process in Experiment 3.

4.1 Defining a Series of Momentum Turning Points

Given the anomaly-corrected, corporate action-adjusted closing price series for each stock, which we obtained in the previous chapter, we are now in a position to try to identify where momentum turning points occur in each price series. To do this, it is essential to first establish a clear definition of momentum itself.

Momentum in financial markets refers to the speed at which a stock's price rises or falls, as well as the duration (strength) of that movement. It is a key indicator which can be used to gauge the strength and direction of a trend. Given this definition, we can identify the times at which there is a significant change in direction of momentum, resulting in a new, persistent trend that follows. Turning points are pivotal in understanding the dynamics of price movements and predicting future price behaviour.

4.1.1 Price Turning Points vs Momentum Turning Points

It is crucial to distinguish between price and momentum turning points since they represent different aspects of market behavior and serve distinct purposes in understanding price dynamics.

Price turning points refer to the moments in time when the direction of a stock's price changes, either from a price increase to a price decrease or vice versa. These are all the peaks and troughs in a price chart. They are straightforward indicators of when a short-term trend in the market has shifted. A momentum turning point, on the other hand, occurs when there is a detected shift in the general long-term price trend, even if the price itself may not reflect a clear turning point. As indicated in **Figure 4.1**, the momentum turning points (green crosses) do not necessarily align with the price turning points (red dots).

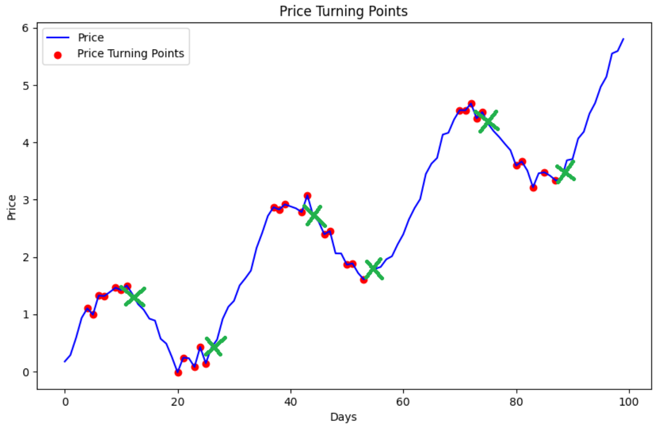


Figure 4.1: Illustration of Price Turning Points in Red and Momentum Turning Points in Green

Note also that, in the illustration, each general uptrend is always followed by a general downtrend, and vice versa. Namely, the turning points are alternating. This motivates the definition/criteria for a series of momentum turning points:

A series of momentum turning points is a sequence of significant alternating local maximums and minimums in an ’appropriately smoothed price series’.

We proceed to quantify what we mean by ’significant’ and ’appropriately smoothed price series’ in the next section where we label these turning points, but for now we note that the ’smoothing’ is carried out to filter the general trend from the noise.

4.2 Labelling Momentum Turning Points

Using the definition that has been established, we must now focus on labelling the price data with momentum turning points. This allows us to train a model, later on, that can classify these turning points on unseen data based on the features derived from the first experiment.

Now, to explain the ’appropriately smoothed price series’ from the previous section, we want to capture the general trend in the price history by smoothing the price data enough so that the noise component of the stocks’ price series can be sufficiently removed whilst still making the overall trend more apparent. For this smoothing, we experiment with moving averages over different rolling window sizes ranging from a length of 5 days to

20 days by leveraging PySpark's Window function, visualising the trend line in each case. An averaging window of 10 days is found to be optimal in balancing noise reduction with trend capture. Based on the predefined criteria for momentum turning points, we proceed by introducing potential algorithms for labelling significant minimums and maximums in the smoothed price series.

4.2.1 Deviation Algorithm Implementation

In this section, we describe the first algorithm considered for labelling momentum turning points, which we refer to as the Deviation Algorithm. The primary objective of this algorithm is to identify specific periods in each stock price history that correspond to significant turning points of the smoothed price. We attempt to achieve this by using a slightly longer-term moving average (compared to the 10-day moving average used to smooth the price) and incorporate a rolling standard deviation of the original smoothed price.

The rationale behind this approach is that maximum and minimum periods are often marked by significant deviations from a longer-term trend[1]. By examining the price relative to its longer-term moving average, and measuring recent price variability through the rolling standard deviation, the algorithm aims to highlight points where the stock price deviates significantly from its average. These deviations would indicate periods of heightened volatility, which may signal potential momentum shifts.

The algorithm is designed to work as follows:

1. Moving Average Calculation: A longer-term moving average is calculated for each stock using PySpark's 'avg' function over a specified rolling window (e.g., 20, 30, or 40 days). This is done by first defining a window to partition the data by the stock identifier and order the data by date within each partition. A rolling window is then created, which slides over each partition to get the moving average series for each stock. The rolling window size is made adjustable for experimentation with different lengths over which the average is calculated.

2. Rolling Standard Deviation: The rolling standard deviation of the smoothed price is calculated using a similar approach, using PySpark's 'stddev' function over a different rolling window, providing a measure of recent variability in price.

3. Once the moving average and standard deviation are computed, the next step is to define the thresholds for significant deviations. When the smoothed price *exceeds* the moving average by a certain multiple of the rolling standard deviation, it is labelled as a potential maximum. Conversely, when it falls *below* the moving average by the same threshold, it is labelled as a potential minimum. Intuitively, the 'when' conditional function in PySpark is used to help carry out the labelling procedure.

To fine-tune the detection of turning points, several combinations of moving average window lengths and rolling standard deviation window lengths were tested. Lengths ranging from 20 days to 60 days were experimented with for both windows, as well as 1 to 3 standard deviations' thresholds for labelling a deviation as a turning point. However, despite the promising theoretical foundation, none of the parameter combinations were able to effectively and consistently pinpoint momentum turning points.

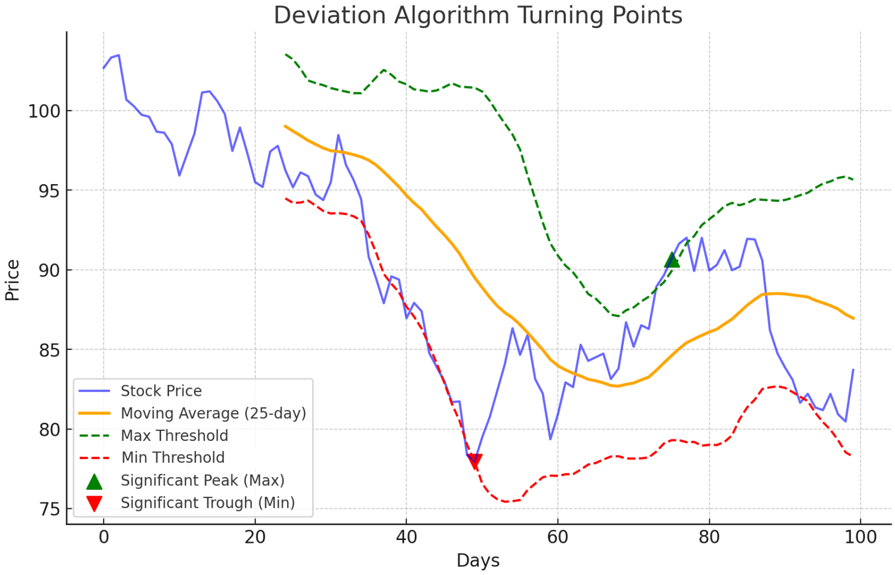


Figure 4.2: Illustration of the Deviation Algorithm's Failure to Capture Significant Peaks and Troughs. The significant peak and trough have been manually labelled.

We visualise an example output of the algorithm, shown in **Figure 4.2**. Notice that the dashed red minimum threshold line intersects the solid blue stock price line at several points, though we only intended for its intersections to occur near the manually labelled significant trough. Although the algorithm performed better in identifying the significant peak, this visualisation illustrates that the algorithm is generally very inconsistent in turning point identification.

The issues encountered with this algorithm are therefore as follows:

- False positives: Many short-term deviations are labelled as turning points, even though they did not correspond to significant shifts in momentum. This was especially problematic when using shorter window sizes.

- Missed turning points: On the other hand, longer window sizes often smoothed out the price data too much, causing the algorithm to miss several legitimate turning points, especially in more volatile stocks.

- Inconsistent results across stocks: The algorithm is highly sensitive to stock-specific volatility, leading to inconsistent performance across different stocks within the same sector.

We therefore adopt a more direct approach for labelling momentum turning points, which better addresses the shortcomings of the previous algorithm. The labels produced by this new algorithm will replace the labels from the previous algorithm for further analysis in the remainder of this study.

4.2.2 Max-Min Windowing Algorithm Implementation

Here we provide the detailed implementation of a more robust algorithm for labelling momentum turning points in the given stock price histories. Having obtained the smoothed price series, we start by finding the minimum and maximum 'smooth price' in every 40 day interval of each stock history using PySpark SQL's built-in 'max' and 'min' functions. For each consecutive block of prices we therefore initially have a maximum label, a minimum label and 38 'regular' labels, which correspond to points that are neither the global minimum nor maximum in the interval. It is important to note that the 40 day interval length was selected after extensive experimentation: a smaller interval tended to produce an excessive number of false positive turning points, while a larger interval risked overlooking significant turning points.

Using the initial daily label series for each stock, we want to identify the *local* minimums and maximums because a subset of these are exactly the turning points that we are looking for. In order to identify these in practice, we begin by considering the boundaries of each 40 day interval, creating a binary column which tells us whether we are at a boundary (taking the value of 1) or not (taking the value of 0). The boundary values for each 40 day interval make up the following sequence: $(1, 0, 0, ..., 0, 1)$ where there are 38 zeroes between the 2 ones (boundaries) on either side (see **Table 4.1**).

Day	**Price ($)**	**Smooth Price ($)**	**Label**	**Boundary**
1	100	101	Regular	1
2	105	102	Regular	0
...	...	...	...	0
19	150	155	Max	0
...	...	...	...	0
40	90	88	Min	1
41	92	89	Regular	1
...	...	...	...	0
63	120	125	Max	0
...	...	...	...	0
74	70	68	Min	0
80	75	70	Regular	1

Table 4.1: Illustration of Boundaries and Initial Labels for Two Consecutive 40-Day Intervals for a Single Stock, where "..." in the Label Column is a Series of 'Regular' Labels

Note that we define two turning points to be 'consecutive' if the the first turning point is followed by the second turning point. However, we define two turning point *labels* to be 'adjacent' if they happen one day after the other. Given these definitions, we carry out the following steps to pinpoint the important local minimums and maximums for each stock, which correspond to the momentum turning points (we use PySpark's coalesce function to join each modified filtered dataframe with the original dataframe, reflecting the changes made in the process. The code for the labelling algorithm can be found in the Appendix section):

1. Replace all adjacent 'min' 'max' labels (or vice versa) with 'regular' 'regular' since these correspond to the global minimum and maximum at the start and end of two consecutive intervals respectively, signifying the middle of a trending period (meaning they are clearly not turning points).

Day	**Label**	**Boundary**	**Replacement Label**
160	Min	1	Regular
161	Max	1	Regular

Table 4.2: Illustration of Min/Max Label Replacement at Adjacent Interval Boundaries

2. Filter the original dataset for labels at the boundaries (where the 'Boundary' column equals 1). If there are two adjacent maximums or two adjacent minimums, replace the higher minimum or the lower maximum with the label 'regular', retaining the more extreme value. Note that this situation corresponds to a local extremum occuring at the boundary. We therefore preserve this by keeping the most pronounced label.

Day	**Smooth Price ($)**	**Label**	**Boundary**	**Replacement Label**
360	100	Min	1	Min
361	101	Min	1	Regular

Table 4.3: Illustration of Min/Min Label Replacement at Adjacent Interval Boundaries

3. Now, filtering the original dataset for only the maximum and minimum labels, if there are two consecutive maximums or two consecutive minimums:

   - If exactly one of the two in the pair is on the boundary, then replace the one on the boundary with 'regular' since it cannot be a turning point (by definition we cannot have two consecutive maximums or two consecutive minimums).

Day	**Label**	**Boundary**	**Replacement Label**
74	Max	0	Max
120	Max	1	Regular

Table 4.4: Illustration of Consecutive Maximum 0-1 Label Replacement

- Else, replace the higher minimum or the lower maximum with 'regular', keeping the more extreme value (again, we cannot have two consecutive maximums or two consecutive minimums, but here we keep the more pronounced label).

Day	**Smooth Price ($)**	**Label**	**Boundary**	**Replacement Label**
98	116	Max	0	Regular
124	142	Max	0	Max

Table 4.5: Illustration of Consecutive Maximum 0-0 Label Replacement

4. Finally, filtering the original dataset again for only the maximum and minimum labels, if consecutive 'min' 'max' labels (or vice versa) occur within 4 weeks of each other with less than a 5% change in smooth price, remove these turning points since they are insignificant.

Day	**Smooth Price ($)**	**Label**	**Replacement Label**
576	116	Max	Regular
590	112	Min	Regular

Table 4.6: Illustration of Insignificant TPs Label Replacement (less than 5% price change in 4 weeks)

We have now implemented a windowing algorithm that thoroughly examines all potential scenarios when trying to identify momentum turning points using min-max analysis on a smoothed price series. Visual analysis of this algorithm's successfulness will be conducted in the Results Section. The price data is now labelled daily with either 'min', 'max' or 'regular', where each 'min' and 'max' corresponds to a minimum and maximum momentum turning point respectively.

4.3 Summary

This chapter outlines a methodology for systematically identifying and labelling momentum turning points in stock price series. It begins by establishing the theoretical foundation for momentum turning points, distinguishing them from price turning points, and then presents two algorithms designed to label momentum turning points in the price data. We pick the algorithm that produces higher-quality labels for further usage in this study.

1. *Defining a Series of Momentum Turning Points*: The chapter begins by clearly defining the concept of momentum in financial markets, explaining its importance in detecting significant changes in price trends. It differentiates between price turning points, which reflect short-term shifts in price direction, and momentum turning points, which indicate more sustained changes in the underlying trend. The section concludes by succinctly defining a series of momentum turning points using the concept of smoothed prices.

2. *Labelling Momentum Turning Points*: Building on the definitions, the chapter details two potential methodologies for labelling momentum turning points in the stock price data, namely the Deviation Algorithm and the Max-Min Windowing Algorithm. The Deviation Algorithm suggests using a moving average and rolling standard deviation to establish thresholds that signal a turning point when surpassed. However, due to several challenges encountered, this approach was ultimately discarded for further analysis. We adopted a more robust approach in the form of the Max-Min Windowing Algorithm. This method involves systematically analysing turning point labels by examining each stock's smoothed price history within 40-day intervals. The interval length was carefully chosen after extensive experimentation to balance the risk of overlooking significant turning points with the need to avoid generating excessive false positives. Through a series of methodical steps, the Windowing Algorithm identifies local minimums and maximums as potential turning points and refines these labels based on boundary conditions to improve accuracy, making it a more reliable solution for further analysis.

The outcome of this chapter is a robust algorithm that labels the stock price data with 'min', 'max', or 'regular' for each day, pinpointing the minimum and maximum momentum turning points. These high-quality labels are vital for the subsequent modelling process in Experiment 3.

Chapter 5

Experiment 3 - Identifying Key Momentum Drivers and Predicting Turning Points

This chapter utilises the engineered feature space from Experiment 1 and the high-quality turning point labels generated in Experiment 2 to produce predictions on whether a momentum turning point will occur within the next four weeks, and if so, what type. The process begins by reducing the original feature space, using statistical models and analyses to identify key momentum drivers. The chapter proceeds by appropriately aggregating the labels, combining them with the features, and then applying classification algorithms to generate the final predictions. It concludes by discussing evaluation metrics and a summary of the modelling process.

5.1 Practical Considerations

When preparing for rigorous data analysis, several practical considerations must be addressed, in particular given the size and complexity of the dataset. Each stock in our dataset has approximately 1,100 weekly observations, and with hundreds of stocks in a single sector, each described by hundreds features, the dataset quickly scales to hundreds of millions of entries for just one sector. Given that our analysis extends across all 19 market sectors, the volume of data becomes enormous, posing significant computational challenges.

Even when leveraging PySpark's repartition function to enable parallel processing, the sheer scale of the dataset means that the computational burden will still be substantial. The repartition function helps distribute the data across multiple nodes, optimising the processing time, but the process still needs to be repeated across all 19 sectors. This will inevitably be very time-consuming, potentially leading to bottlenecks in the data processing pipeline, especially when dealing with limited computational resources and working within tight time constraints.

To mitigate these challenges and ensure the feasibility of the feature selection and model training process, we propose a sampling strategy. Instead of training on the entire dataset for each sector, we will randomly select a subset of 5 stocks from each sector. This approach dramatically reduces the size of the dataset while still maintaining the diversity of data necessary for robust model training. By focusing on a smaller, representative sample, we can expedite the training process and manage computational resources much more effectively.

This strategy also still results in separate models being trained for each market sector, ensuring that the unique characteristics and behaviours of stocks within each sector are captured, potentially leading to accurate predictions. While this approach reduces the overall data volume, it retains the critical sector-specific information needed to build effective models. The dataset for each individual sector now contains fewer than 5 million entries.

5.2 Feature Selection

The ultimate goal of most trading strategies is to maximise returns as much as possible, and this is also true for our case, though indirectly, since we are trying to produce a signal that can be used in these strategies to maximise returns. Naturally we want to try to find the best predictors of returns using a rigorous feature selection procedure. In doing so, we identify the key momentum drivers, which will be used as input to our prediction models. The features removed and retained at each stage of the selection process is detailed in the Appendix section.

5.2.1 Structuring the Data for Feature Selection

Given the adjusted price series, which we calculated in experiment 1 for each stock, we compute the monthly returns (looking backwards) corresponding to each price time point using the formula:

$$\text{monthly return} = \frac{(\text{adj. price now}) - (\text{adj. price 1 month ago})}{\text{adj. price 1 month ago}}$$

where in our calculation we assume each month is 20 business days long. We can then shift these returns back by one month to get monthly forward returns at each price time point, which is exactly the target variable that we are trying to maximise.

Now, it's important to note that we have two datasets: one containing feature sets aligned with weekly earnings dates, and another with next-month forward returns for each day in the stock histories. Given this, we perform a left-join of the returns on the features, creating a combined dataframe that includes the one-month forward return alongside the corresponding set of feature values for each weekly reporting date, organised in a time series for each stock in our sample. We have now structured the data to be well-suited for the upcoming feature selection process.

5.2.2 Removing Static and Low Variance Features

Before we apply a feature importance model, we must first eliminate features which do not have significant variability, because these do not contribute much information. Retaining such features can introduce noise, thereby reducing the performance of our model.

Given our random sample of 5 stocks from the current sector, we analyse the variance of the features in this sample. We set a variance threshold to determine which features should be removed. If the variance of a feature is below this threshold for more than half of the stocks in the sample, we deem the feature to be static or of low variance and consequently drop it from the dataset. We find a variance threshold of 0.1 to be optimal for this process, resulting in around 50-100 features being dropped, dependent on the sample from that sector. This leaves us with a feature space that consists of around 225 features for each sector, which we try to narrow down further through feature importance analysis.

5.2.3 Finding the Most Important Features

Reducing the number of features can lead to more interpretable models, faster training times, and potentially improved model performance due to the elimination of irrelevant or redundant information. In this study, we begin by employing a Random Forest (RF)-based feature selection method to pick the top features out of the ones that are left. Following this, we drop features from correlated feature pairs based on their RF importances, and remove features that do not share any mutual information with returns.

Random Forest Feature Importance

Since our data is now relatively compact, using Pandas is likely more efficient and faster than Spark. This is because Pandas processes data locally, eliminating the extra complexity and time involved in distributing tasks across multiple nodes in the Spark cluster. We begin by separating the features from the target variable (returns). Using the 'RandomForestRegressor' from scikit-learn, we apply a grid search for hyperparameter tuning, leveraging 'TimeSeriesSplit' to ensure time-ordering in the cross-validation process. Specifically, we use 9-split rolling-forward cross-validation ('n_splits' = 9; **Table 5.2**) to select the optimal model configuration that minimises the average root mean square error (RMSE) across all validation folds. Using 9 splits offers a good trade-off between bias, variance, and computational cost. It ensures that each fold has enough data to give meaningful performance evaluations without being too computationally expensive. The grid search tunes four key hyperparameters for the random forest:

- max_features, which controls the number of features considered when splitting nodes in each decision tree
- max_depth, which limits the maximum depth of each tree
- min_samples_split, which controls the minimum number of samples required to split an internal node, and
- min_samples_leaf, which sets the minimum number of samples that a leaf node must have

Note that 'n_estimators', which represents the number of trees in the forest, is not tuned for our models. Simply setting 'n_estimators' to a sufficiently large default (e.g. the total number of features at this stage) can achieve good out-of-sample performance without requiring fine-tuning, saving time and computational resources. We omit the analysis of the individual folds in this study for the sake of brevity, however, most importantly, we obtain optimal values for the hyperparameters, as shown in **Table 5.1**. The remaining random forest hyperparameters are left at their default values, as they have minimal influence on the average RMSE.

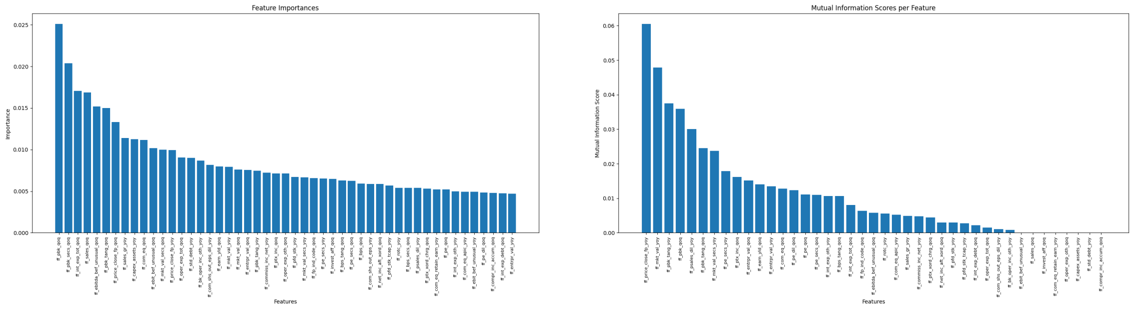
Hyperparameter	**Range of values**	**Optimal value**
n_estimators	{225}	225 - no tuning required
max_features	{auto, sqrt}	sqrt
max_depth	{3, 5, 7, 10}	5
min_samples_split	{2, 5, 10, 15, 20}	2
min_samples_leaf	{1, 2, 4, 8, 16}	1

Table 5.1: Optimised Hyperparameters for Random Forest Feature Importance Model

After training the Random Forest, we extract the importance of each feature, which indicates the contribution of each feature to the accuracy of return predictions. We proceed by selecting the top quintile of features according to the model, giving us around 45 features in total at this stage of the selection process (see **Figure 5.1a**).

Dropping Features Using Correlation Analysis

The next stage of our dimensionality reduction process is to identify features that we want to remove from highly correlated feature pairs. In the presence of multicollinearity, it becomes difficult to determine the effect of each predictor variable on the response variable because changes in one predictor may be associated with changes in another predictor. It can also indicate that some features can be removed without losing significant information. We use Pearson's Correlation Coefficient to measure the linear relationship strength between each pair of features, X and Y, defined mathematically as:



(a) Top Quintile of Features Selected from RF Regression Against Returns

(b) Aftermath of Correlation Analysis: MI Shared with Returns for Remaining Features

Figure 5.1: Random Forest Importances Before Correlated Feature Removal; Mutual Information Scores After Correlated Feature Removal - Finance Sector

$$\rho_{XY} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

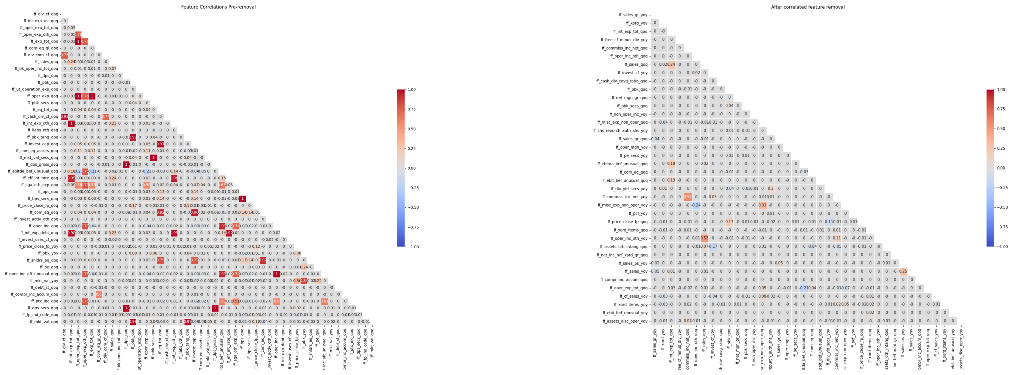
where X_i and Y_i are the individual values of features X and Y for the i-th observation, $\bar{X}$ and $\bar{Y}$ are the means of features X and Y respectively, and n is the number of observations. ρ_{XY} can range from -1 (perfect negative correlation) to 1 (perfect positive correlation), with 0 indicating no linear correlation.

We examine our random sample of 5 stocks in the sector to determine the features to remove due to correlations. We calculate Pearson's correlation between each pair of features to obtain a correlation matrix for each stock in the sample. We then take an average of the 5 correlation matrices to obtain a matrix of average correlations between each pair of features. From this, we identify correlations greater than a threshold of 0.7, and eliminate the features in these highly correlated pairs (**Figure 5.2**) with the lower random forest feature importance. Overall, this process removes around 10 additional features, leaving us with roughly 35 features at the end of this stage for each sector.

Removing Non-Informative Features

Given the steps already taken to reduce the feature space for our study, it is essential to further refine our feature selection process to ensure that we have the most effective set of predictors for stock returns. RF feature importance and correlation analysis have already provided significant benefits in ensuring diversity among the features. However, to further enhance the predictive power of our model, we must also consider the relevance of each feature in terms of its mutual information (MI) with the returns target variable.

Using the scores that we obtain from MI regression, we can identify and drop features that do not provide unique information about returns. We leverage the 'mutual_info_regression' function from scikit-learn to extracting the mutual information scores for each feature. We remove those that have a score of zero, thereby retaining only those features that con-



(a) Correlation Matrix Before Feature Removal (b) Correlation Matrix After Feature Removal

Figure 5.2: Comparison of Correlation Matrices Before and After Feature Selection - Finance Sector

tribute significantly to predicting stock returns. In total, this final stage of the feature selection process leaves roughly 30 features, per sector, to use as input to our subsequent models (the features with non-zero MI scores in **Figure 5.1b**)).

5.3 Structuring the Data for Class Prediction

We now shift our focus to structuring the data to enable accurate and meaningful predictions of momentum turning points. The data from the first two experiments provides the foundation for this task, namely the weekly sets of sector-relative features obtained from

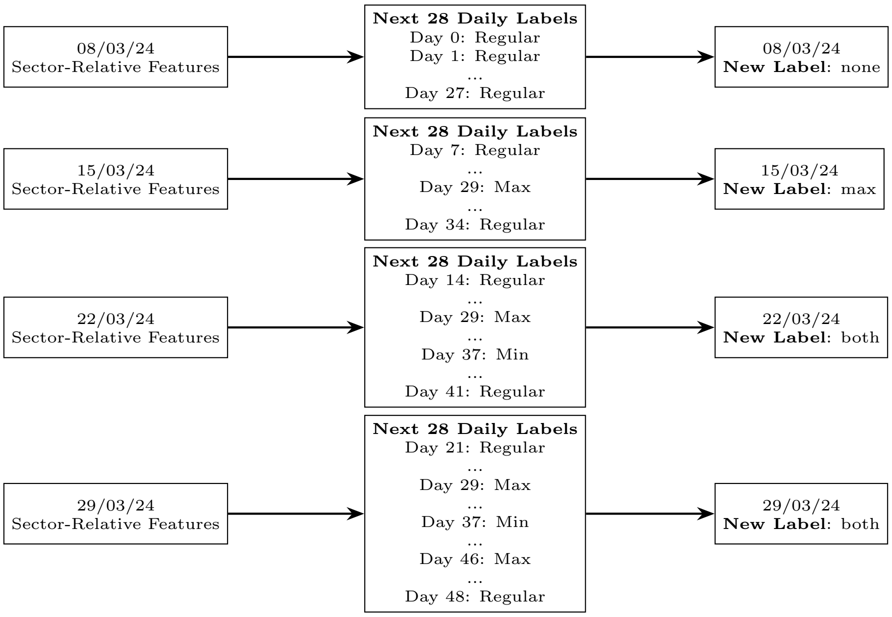


Figure 5.3: Illustration of Conversion to Weekly Labels for a Single Stock. Each '...' represents intermediate days with 'Regular' labels. Each new label reflects the class of momentum turning point(s) over the following four weeks. Day γ denotes γ days after 08/03/24.

the first experiment, which have now been reduced to key momentum drivers, and the daily turning point labels derived from the second experiment. These two datasets, however, differ in their frequency since the features are recorded weekly, while the turning point labels are at a daily resolution. To effectively combine these datasets for predictive modelling, we must first align their frequencies. Note that from this point forward, the terms "features" and "key momentum drivers" will be used interchangeably.

The turning point labels from Experiment 2 indicate whether a point in time corresponds to a maximum turning point ('max'), a minimum turning point ('min'), or a regular point that is neither a maximum nor a minimum ('regular'). Since our features are measured on a weekly basis, it is necessary to aggregate the daily turning point labels to match this weekly frequency. To achieve this, we consider adjusting the labels so that they indicate, in the next four weeks from each feature record date, whether there is a minimum ('min'), maximum ('max'), no turning point ('none'), or whether both a minimum and a maximum ('both') will occur (**Figure 5.3**).

We must now justify the possibility of both a minimum and maximum momentum turning point occurring within the same month by revisiting the algorithm from Experiment 2. Recall that our labeling process only excluded consecutive turning points that occurred within four weeks of each other *and* had less than a 5% change in price. Consequently, consecutive ('min', 'max') pairs that occur within the same four-week period with a price change *greater* than 5% likely remain intact. Therefore, it is appropriate to introduce a separate label, 'both', to account for such periods.

Given that the features and the labels are now synchronised, we proceed to seamlessly combine the dataframes on the common weekly dates for each stock. We ensure that each set of features corresponds directly to the correct label for the same week, resulting in a dataset that serves as the foundation for building our multi-class prediction models.

5.4 Predicting Turning Points

The features and the turning point labels are now in an appropriate format, ready for generating predictions using two ensemble learning models: the Random Forest and Gradient Boosting Machine (GBM) multi-class classification models. The Random Forest model minimises the Gini impurity when splitting nodes in decision trees, whereas the GBM minimises a multinomial log loss objective through a softmax transformation. These models will allow us to predict whether a turning point will occur, and if so, whether it will be a maximum, minimum or both.

5.4.1 Calculating Class Weights

One of the key challenges we face is the imbalance in the distribution of the target labels, as we expect turning point periods, periods with either maximums or minimums, to be relatively infrequent compared to no turning point periods (periods without any turning points). To handle this imbalance, we use the 'compute_class_weight' utility from sklearn.utils, which calculates the class weights automatically based on the distribution of the target classes. This method for calculating class weights is based on the inverse of the class frequencies:

$$w_c = \frac{n}{k \cdot f_c}$$

where:

- w_c is the weight for class c,
- n is the total number of samples,
- k is the number of classes, equal to 4 in our case
- f_c is the frequency of class c (i.e., the number of samples belonging to class c).

We ensure that minority classes (e.g. 'both', where both turning points occur in the next period) are given more importance during model training, thereby reducing the risk of the model being biased towards the majority classes. To simplify the training process, we compute the average class weights across all training folds. The computed class weights are then returned as an array, which we convert into a dictionary to pass into the 'RandomForestClassifier' and 'GradientBoostingClassifier' models.

5.4.2 Cross-Validation and Evaluation

As we did for the RF feature importance model, since we are working with time-series data, we apply 9-split rolling-forward cross-validation to evaluate the performance of each model on unseen data while preserving the temporal structure of the dataset.

Again applying grid search for hyperparameter tuning, we specify the parameter grid for the following hyperparameters (as before, we set n_estimators, the number of trees, equal to the number of features at this stage):

- For Random Forest Classifier: max_features, max_depth, min_samples_split, min_samples_leaf
- For GBM, in addition to the above, we tune the learning_rate, which controls the contribution of each tree to the overall model.

Given the imbalance in our dataset, we average the *weighted* F1 score across all validation folds, illustrated in **Table 5.2** by each couple of 'O's, to evaluate our current hyperparameter configuration, ensuring that the performance on the minority classes is adequately accounted for. Finally, we fit the 'GridSearchCV' object to the training data and find the best performing model for each of the Random Forest and Gradient Boosting

Train-validation	Years											
	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011
Split 1	X	X	O	O								
Split 2	X	X	X	X	O	O						
Split 3	X	X	X	X	X	X	O	O				
Split 4	X	X	X	X	X	X	X	X	O	O		
Split 5	X	X	X	X	X	X	X	X	X	X	O	O
Split 6	X	X	X	X	X	X	X	X	X	X	X	X
Split 7	X	X	X	X	X	X	X	X	X	X	X	X
Split 8	X	X	X	X	X	X	X	X	X	X	X	X
Split 9	X	X	X	X	X	X	X	X	X	X	X	X

Train-validation	Years												
	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	
Split 1									*	*	*	*	*
Split 2									*	*	*	*	*
Split 3									*	*	*	*	*
Split 4									*	*	*	*	*
Split 5									*	*	*	*	*
Split 6	O	O							*	*	*	*	*
Split 7	X	X	O	O					*	*	*	*	*
Split 8	X	X	X	X	O	O			*	*	*	*	*
Split 9	X	X	X	X	X	X	O	O	*	*	*	*	*

Table 5.2: Train-Validation Splits for the Years 2000-2024. The first table covers the years 2000-2011, while the second table covers 2012-2024, with 'X' indicating years used for training, 'O' for validation, and '*' for testing.

classifiers. We use the best models to make predictions on the test dataset (marked by the '*'s in **Table 5.2**). These predictions are then evaluated using the 'classification_report' function from scikit-learn, which provides detailed performance metrics including the precision, recall, and F1 score for each class ('min', 'max', 'none' and 'both').

5.5 Summary

This chapter presents a systematic approach to identifying and predicting momentum turning points in stock price data using the engineered features from Experiment 1 and the turning point labels from Experiment 2. The primary goal is to produce weekly predictions indicating whether a momentum turning point will occur within the next four weeks and if so, its type. The chapter outlines practical considerations for data handling, details the feature selection process, and explains the model-building approach for turning point predictions. The following key aspects are covered in this chapter:

1. *Practical Considerations:* The data spans 24 years, consisting of approximately 1,100 weekly observations for each stock, with hundreds of stocks per sector and hundreds features per stock. This results in a vast dataset even for a single sector, posing significant computational challenges. Even with PySpark's repartition function for parallel processing, handling the entire datasets across all 19 market sectors is time-consuming. To mitigate this, we propose selecting a random subset of 5 stocks per sector for analysis and model training, reducing the overall dataset size while retaining sector-specific characteristics.

2. *Feature Selection*: The chapter defines the target variable as the next-month's (4 week) forward returns, calculated from the adjusted prices obtained in Experiment 1. We align this dataset with weekly feature sets, creating a dataset structured for feature selection. The feature selection process starts by removing static and low variance features, where a variance threshold of 0.1 eliminates around 50-100 features. The remaining 225 features are then further refined using a Random Forest (RF)-based feature selection method, followed by correlation analysis. Features with high correlations (above 0.7) are removed, leaving approximately 35 features. Lastly, mutual information (MI) regression is applied to ensure that only the most informative features remain, reducing the final feature set to around 30 features per sector, which will be used for the prediction models.

3. *Structuring the Data for Class Prediction*: The features and turning point labels differ in their frequency, with features recorded weekly and labels recorded daily. To align them, the daily turning point labels are aggregated to match the weekly feature frequency, resulting in a unified dataset where each set of features corresponds to a label indicating whether a momentum turning point (maximum, minimum, both, or none) will occur in the next four weeks.

4. *Turning Point Predictions*: With the features and labels now synchronised, we proceed to build two multi-class classification models: Random Forest and Gradient Boosting Machine (GBM). Given the imbalance in the target labels, class weights for the models are adjusted based on the inverse class frequency. Rolling forward cross-validation is used, ensuring that the temporal order of the data is preserved. Grid search is applied for hyperparameter tuning and the weighted F1 score is used as the performance metric on the validation folds to account for the imbalanced class distribution. After identifying the best performing models, predictions are evaluated through precision, recall, and F1 scores for each class.

This chapter details the development of two robust tree-based models designed to provide weekly predictions of momentum turning points for the following month. The chapter blends practical methodologies with their theoretical foundations to smoothly transition from research design to empirical execution, demonstrating how conceptual insights are applied in practice, resulting in the successful implementation of predictive models.

Chapter 6

Results

This chapter presents the findings of the analyses conducted across the three experiments, providing a detailed explanation of the results obtained at each stage of the ML pipeline used in the study. It offers insights into the key drivers of momentum identified and the performance of the predictive models. The chapter concludes with a comprehensive comparison between the classification methodologies, evaluating their effectiveness in predicting momentum turning points.

6.1 Engineered Features

We begin by revisiting the two distinct approaches that were explored in this study to generate features for predicting momentum turning points. The first approach focused on rate-of-change transformations, where the goal was to capture the quarterly and annual changes in key financial metrics. However, after initial testing, this approach was ultimately deemed inadequate due to its lower frequency of updates. The second approach, which proved more effective, was the sector-relative feature engineering method.

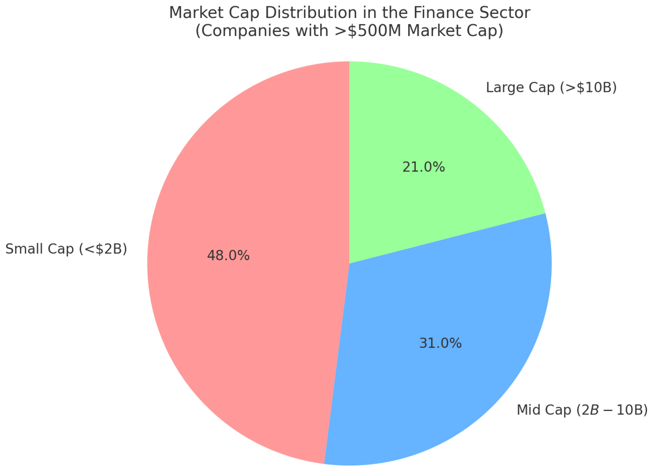


Figure 6.1: Distribution of Market Cap Categories in the Finance Sector

Here, we aimed to create features that measured the relative performance of a company's fundamentals on a weekly basis, directly comparing each company's financial metrics to those of its peers within the same market value category in the same sector. An example distribution of the companies' market cap classes is shown for the Finance Sector in **Figure 6.1**. By calculating sector-relative averages for each financial metric, we were able to create features that represented how a company's performance deviated from the average of its sector in real time. This method provided higher-frequency data, resulting in more sensitive changes in the company fundamentals and aligning better with the needs of momentum analysis. We display the output of the engineering procedure for the 'sales' feature and 'price-to-book ratio' feature for a small subset of our entire dataframe in **Table 6.1**.

fsym	**date**	**ff_sales feature (2dp)**	**ff_pbk feature (2dp)**
C0TNSZ-S	2000-03-24	8530.00	3.04
C0TNSZ-S	2000-04-07	-1254.61	0.00
C0TNSZ-S	2000-04-14	-732.02	1.19
C0TNSZ-S	2000-04-21	-1165.42	1.06
C0TNSZ-S	2000-04-28	-1136.05	6.63
F5YZB7-S	2000-03-24	9198.74	3.23
F5YZB7-S	2000-04-07	-585.87	0.00
F5YZB7-S	2000-04-14	-63.28	1.38
F5YZB7-S	2000-04-21	-496.68	1.26
F5YZB7-S	2000-04-28	-467.31	6.82
HVQ6TN-S	2000-03-24	9582.76	1.90
HVQ6TN-S	2000-04-07	-201.85	0.00
HVQ6TN-S	2000-04-14	320.74	0.05
HVQ6TN-S	2000-04-21	-112.67	-0.08
HVQ6TN-S	2000-04-28	-83.30	5.49

Table 6.1: Filtered Dataframe Showing Engineered Feature Time-Series for ff_sales and ff_pbk for 3 Stocks from the Finance Sector from End of March to End of April 2000. Note that the 'fsym' column contains the stock identifier.

Each row in the table corresponds to a specific stock and date, with the stocks stacked on top of each other according to the given date range (for the entire dataframe, this is generalised to complete stock histories being stacked on top of each other). From here on, when we refer to a column e.g. 'ff_sales' or 'ff_pbk' we mean the stacked feature series corresponding to that column.

The variability of feature values is apparent in the dataset. For example 'ff_sales' shows fluctuations between large positive and negative values. Positive values reflect periods of increased sales relative to similar companies, while negative values may indicate a relative downturn in sales due to financial pressure. On the other hand, 'ff_pbk', measuring the relative price to book ratio, indicates whether stock price, compared to its book value, is better (higher) or worse (lower) compared to other similar companies at each point in time. Each observation in our dataframe therefore holds key information about a company's past performance, and it is highly likely that a subset of these features can be used to make predictions about future stock performance.

Note that most zero values in the dataframe (noticeable in 'ff_pbk' above) are the result of filling missing values after feature engineering. The zeros here were used as a proxy for sector average values since they indicate no difference to the average.

6.2 Turning Point Labels

Recall that we explored two different algorithms for labelling momentum turning points. The first, the deviation algorithm, aimed to identify turning points by using a moving average combined with a rolling standard deviation to set dynamic thresholds. The idea was that when the stock price exceeded the thresholds, it would indicate a potential turning point. However, the deviation algorithm was ultimately discarded due to challenges in finding optimal window sizes and thresholds. The second approach, the max-min windowing algorithm, offered a more direct and robust solution. This method involved analysing the smoothed price series within predefined 40-day intervals and identifying the local minimums and maximums as turning points. By systematically reviewing the stock price history within these intervals, the algorithm was able to more effectively capture significant turning points while minimising false signals, as will be demonstrated in the upcoming visualisations.

We begin by examining the distribution of the daily label series for a few different sectors, which consists of three possible labels: 'max', 'min', or 'regular'. This initial step is taken to verify the frequency — or rather, the infrequency — of turning points in the dataset. We analyse the Finance, Commercial Services and Technology Services sectors given that they are relatively large sectors with diverse market dynamics which represent a broad cross-section of the economy.

Label	**Finance**	**Commercial Services**	**Technology Services**
Regular	95.80%	95.74%	96.24%
Max	2.10%	2.13%	1.88%
Min	2.10%	2.13%	1.88%

Table 6.2: Distribution of Daily Turning Point Labels for the Finance, Commercial Services, and Technology Services Sectors

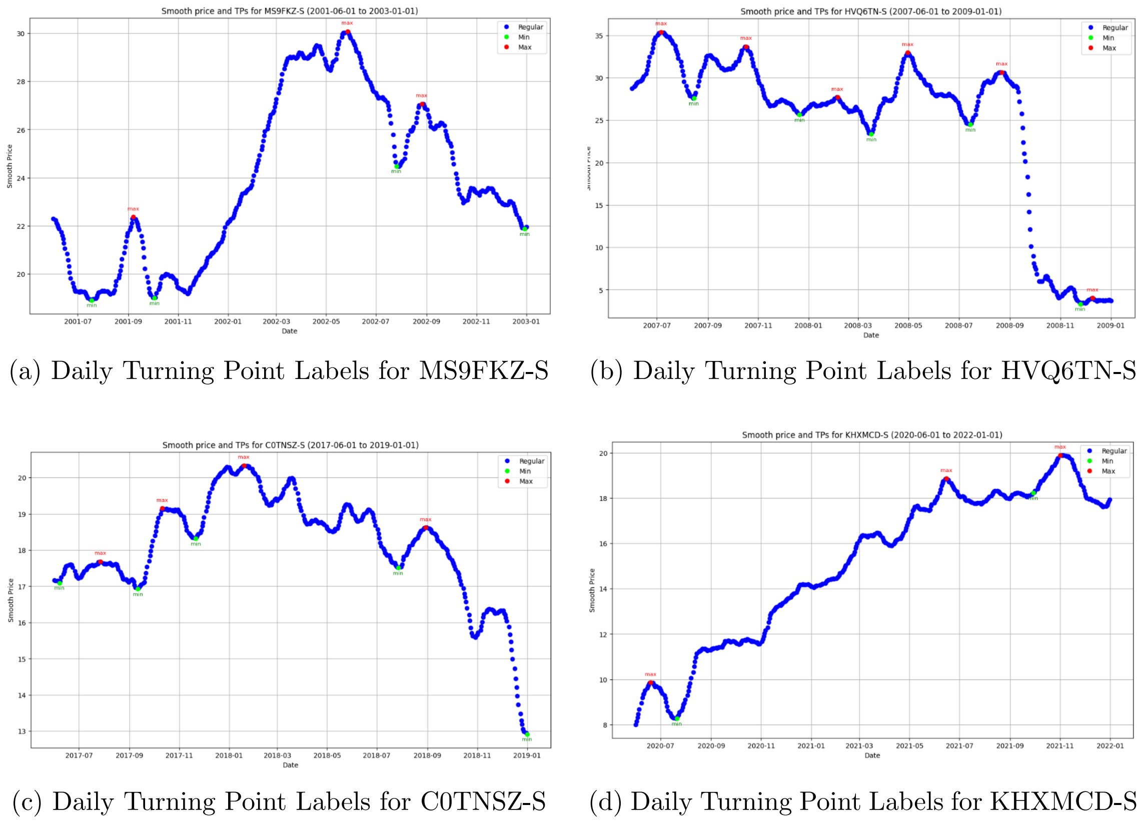
The results of the label distribution analysis, as seen in **Table 6.2**, show that the majority of the dataset consists of 'regular' points, making up around 96% of the total labels for each sector. This aligns with expectations, as momentum turning points (both 'max' and 'min') are relatively rare occurrences. Both the maximum and minimum turning points individually account for around 2% of the total labels. The balance between 'max' and 'min' labels is also expected, given the alternating nature of a series of turning points.

The proportions of maximums and minimums each being below 2.5% further align with the algorithm's design, which, by construction, ensures that there is at most one maximum and one minimum within each 40-day window.

While this provides a level of confidence that the algorithm is working as intended, it does not fully confirm its correctness. These results demonstrate that the algorithm is behaving as expected in terms of frequency distribution, but it is essential to validate the turning point locations with a more qualitative assessment. To achieve this, we will proceed by visualising the labelled turning points for a sample of stocks from the Finance sector. These visualisations will help confirm that the algorithm is effectively capturing both maximum and minimum momentum turning points while filtering out false positives.

6.2.1 Graphical Validation of the Max-Min Windowing Algorithm

We now proceed to validate our algorithm by visually inspecting label-colour-coded price charts for a random sample of stocks during different 1.5 year time periods. This timeframe strikes a balance by preventing the graph from becoming overly congested with maximum and minimum labels, which might happen when viewing longer periods, while also ensuring that turning points do not appear too sparsely distributed, as could occur when viewing shorter periods.



(a) Daily Turning Point Labels for MS9FKZ-S

(b) Daily Turning Point Labels for HVQ6TN-S

(c) Daily Turning Point Labels for C0TNSZ-S

(d) Daily Turning Point Labels for KHXMCD-S

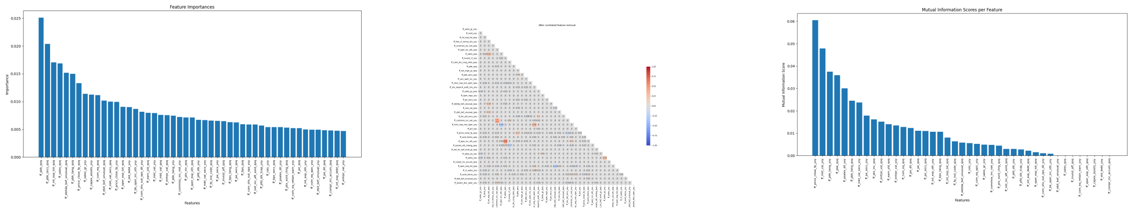
Figure 6.2: Maximum and Minimum Momentum Turning Points for Various Stocks

The graphs presented above illustrate the algorithm's accuracy in labelling momentum turning points, with significant peaks marked as 'max' (red) and significant troughs marked as 'min' (green) in the smoothed price series. These turning points occur in between major inflection points in the stock prices, effectively marking the initiation of new long-term trend directions. This aligns with the theory of momentum turning points, where a shift in price momentum signals the beginning of a sustained trend.

In all graphs, the turning points alternate as expected, ensuring that each uptrend is followed by a downtrend and vice versa, adhering to the definition of a series of momentum turning points. The algorithm avoids false positives by filtering out smaller, short-lived fluctuations, capturing only the significant momentum shifts. The consistent alignment with the major peaks and troughs across different stocks highlights the robustness and reliability of the max-min windowing approach.

6.3 Key Momentum Drivers

After labelling the turning points in the price data, we focused on narrowing down the set of potential momentum drivers using a 4 week forward returns the target variable. This was achieved through a multi-step feature selection process. First, we applied variance thresholding to remove features with low variability that offered little predictive value.



(a) Top Quintile of Features from RF Regression on Sample

(b) Correlation Matrix After Feature Removal

(c) MI Scores After Correlation Analysis on Sample

Feature	Importance
ff_pbk	0.025106
ff_pbk_secs	0.020371
ff_int_exp_tot	0.017044
ff_sales	0.016888
ff_ebitda_bef_unusual	0.015164
ff_pbk_tang	0.014976
ff_price_close_fp	0.013299
ff_sales_gr	0.011390
ff_capex_assets	0.011263
ff_com_eq	0.011172

(d) Top 10 Important Features from RF Regression

Features Dropped (Corr)
ff_bps_secs
ff_price_close_fp
ff_mkt_val
ff_bps
ff_com_shs_out_eps
ff_pbk_secs
ff_ebit_bef_unusual
ff_mkt_val_secs

(e) The 8 Features Dropped due to Correlation Analysis

Features Dropped (MI)
ff_sales
ff_invest_aff
ff_com_eq_retain_earn
ff_oper_exp_oth
ff_capex_assets
ff_std_debt
ff_compr_inc_accum

(f) The 7 Features Dropped due to Mutual Information Analysis

Figure 6.3: Summary of Feature Selection and Removal for the Finance Sector. Graphs (a), (b), (c) correspond to tables (d), (e), (f) respectively.

Next, we used a random forest feature importance model to rank features based on their contribution to model accuracy. We then performed correlation analysis to eliminate redundant features from highly correlated pairs. Finally, features that did not share mutual information with the 4 week forward returns were removed, leaving only those most relevant to predicting momentum shifts: we define these to be our 'key momentum drivers'. We present the most prominent outcomes of the analyses (**Figure 6.3**). A full list of the features removed at each stage of the pipeline (including during preprocessing) can be found in the Appendix section. Note that here we present the results for a single sector for clarity, though these drivers were found for all sectors.

As we can see, different analytical methods remove features based on their specific criteria, meaning that certain features may be removed by one type of analysis but retained by another. For instance, 'ff_sales', which was considered important by the random forest regression, was later removed during the mutual information analysis because it did not provide enough unique information about the target variable. This was an essential aspect of refining the selection process, as each technique had a unique perspective on what constitutes an important or redundant feature. Finally, we proceed to list the features that were found to be the 'key momentum drivers' given our extensive analyses.

Key Momentum Drivers			
ff_pbk	ff_earn_yld	ff_pe_secs	ff_pe
ff_int_exp_tot	ff_mkt_val	ff_bps_tang	ff_int_exp_oth
ff_ebitda_bef_unusual	ff_entrpr_val	ff_net_inc_aft_xord	ff_com_eq_apic
ff_com_shs_out_eps_dil	ff_pbk_tang	ff_pfd_stk_tcap	ff_int_exp_debt
ff_sales_gr	ff_commiss_inc_net	ff_rotc	
ff_com_eq	ff_ptx_inc	ff_psales_dil	
ff_price_close_fp	ff_pfd_stk	ff_ptx_xord_chrg	
ff_oper_exp_tot	ff_mkt_val_secs	ff_ebit_bef_unusual	
ff_bk_oper_inc_oth	ff_fp_ind_code	ff_pe_dil	

Table 6.3: The 31 Key Momentum Drivers for the Finance Sector

A thorough description of these key momentum drivers is provided in the Appendix. The sector-relative features identified in this study as key momentum drivers capture a range of financial metrics, reflecting a company's growth potential, profitability, capital structure, and market valuation relative to its peers. They play a crucial role in determining momentum in stock prices.

For example, market valuation features such as market value, 'ff_mkt_val', and enterprise value, 'ff_entrpr_val', provide a snapshot of how the market perceives the company relative to its competitors. Changes in these metrics can serve as momentum signals by indicating shifts in market sentiment or broader economic factors impacting stock prices. Features related to income and expenses, such as 'ff_int_exp_tot' (total interest expense) and 'ff_oper_exp_tot' (total operating expenses), provide insights into the financial burden on a company. High interest expenses can decrease profitability and reduce upward momentum, while lower operating expenses indicate that a company is managing its costs more effectively compared to its sector peers, potentially boosting stock performance.

While key momentum drivers such as earnings yield and price-to-book ratios are important across all sectors due to their focus on profitability and valuation, certain features are more informative in particular sectors. For instance, 'ff_cogs' (cost of goods sold) plays a crucial role in manufacturing and retail, where production efficiency directly affects profitability. Likewise, 'ff_rd_exp' (R&D expenses) is especially significant in technology and pharmaceuticals, where innovation drives growth. Such factors contribute to the differences in performance of our classification models across various sectors.

6.4 Class Predictions

We move forward with the analysis by discussing the outcomes of the Random Forest (RF) and Gradient Boosting Machine (GBM) models for class predictions. Having already identified the key momentum drivers (most important features) for each sector, and aggregated the daily labels into weekly categories of 'max', 'min', 'none', or 'both', describing the next four weeks, we now evaluate how well these models perform in their weekly predictions.

6.4.1 Class Weights and Hyperparameter Optimisation

We begin by examining the average class weights (or more specifically the average inverse class frequencies) calculated for the 4 different period types. We provide the weights for 3 market sectors out of the total 19 to allow us to easily compare or contrast the class weightings for different sectors.

Sector	**max**	**min**	**both**	**none**
Finance	0.88	0.88	1.80	0.85
Commercial Services	0.94	0.94	1.11	1.00
Technology Services	0.80	0.80	8.49	0.72

Table 6.4: Average Class Weights for 'max', 'min', 'both', and 'none' Categories Across Three Sectors. Weights < 1 give less importance during training. Weights > 1 give more importance during training.

Higher values in the table correspond to less frequent occurrences of certain labels. In the Technology Services sector, the high class weight for "both", 8.49, and low weight for "none", 0.72, are indications of the rarity of both turning point types occurring within the same four week period. This aligns with the sector's susceptibility to trending markets, driven by rapid innovation, frequent disruptions, and global market demand[24]. In contrast, the Finance sector shows a more moderate class weight for "both", 1.8, reflecting a more balanced trend profile, where peaks and troughs occur with greater regularity. This is consistent with the sector's dependence on external factors such as interest rates and regulatory changes, resulting in less persistent trends compared to Technology Services. Commercial Services exhibits the lowest class weight for "both", 1.11, and relatively higher weights for "max" and "min", both 0.94, which is consistent with the sector's typically stable nature. Trends in this sector tend to be driven by factors such as changes in demand for services or cost-cutting initiatives, but they are less likely to result in sustained trends compared to the other two sectors.

Having found the suitable class weights for each sector, we now present the optimal hyperparameters for each model. Recall that we performed grid search tuning using the weighted F1 score, averaging across all validation folds (specified in **Table 5.2**)). As before, we present the results for three sectors, with each set of optimal values corresponding to a model for a specific sector. Since we tune both Random Forest (RF) and Gradient Boosting Classifiers (GBM), this gives a total of 6 models across the 3 sectors.

Random Forest Classifiers - Optimal Hyperparameter Sets by Sector				
Hyperparameter	**Range of Values**	**Finance**	**Commercial Services**	**Technology Services**
n_estimators	{30} - not tuned.	30	30	30
max_features	{auto, sqrt}	sqrt	sqrt	sqrt
max_depth	{3,5,7,10}	3	5	5
min_samples_split	{2, 5, 10, 15, 20}	2	2	2
min_samples_leaf	{1, 2, 4, 8, 16}	1	1	1

Gradient Boosting Classifiers - Optimal Hyperparameter Sets by Sector				
Hyperparameter	**Range of Values**	**Finance**	**Commercial Services**	**Technology Services**
n_estimators	{30} - not tuned.	30	30	30
max_features	{auto, sqrt}	sqrt	sqrt	sqrt
max_depth	{3,5,7,10}	3	3	3
min_samples_split	{2, 5, 10, 15, 20}	2	5	2
min_samples_leaf	{1, 2, 4, 8, 16}	1	1	2
learning_rate	{0.05, 0.1, 0.2}	0.2	0.1	0.1

Table 6.5: Optimal Hyperparameters: Random Forest and Gradient Boosting Classifiers for Finance, Commercial Services and Technology Services Sectors

We are now ready to evaluate the performance of these 'optimal' Random Forest and Gradient Boosting Classifiers on the test set, which consists of data from the beginning of 2020 onwards, as indicated in **Table 5.2**.

6.4.2 Out-of-Sample Evaluation

We evaluate the classification performance metrics for each turning point category individually and each model as a whole (refer to **Table 6.6**).

We begin by defining the concept of 'support' mentioned in the classification report. The support is characterised by the number of actual instances in each class for the test set. The results reveal a noticeable discrepancy in the supports of 'max' and 'min' labels, largely due to the label conversion process and the introduction of the 'both' and 'none' labels. The conversion effectively splits occurrences where both turning points happen within the same period, resulting in a reduced number of pure 'max' and 'min' labels. Despite this, the support numbers for 'max' and 'min' remain fairly similar, which can be attributed to the natural alternation of turning points, as discussed previously.

RF Finance	**Precision**	**Recall**	**F1-score**	**Support**
max	0.35	0.32	0.33	332
min	0.27	0.41	0.33	324
both	0.35	0.31	0.33	161
none	0.36	0.35	0.35	336
Weighted Avg	0.33	0.35	0.34	**Total**: 1153

GBM Finance	**Precision**	**Recall**	**F1-score**	**Support**
max	0.34	0.35	0.34	332
min	0.34	0.39	0.36	324
both	0.35	0.32	0.32	161
none	0.45	0.29	0.35	336
Weighted Avg	0.37	0.34	0.35	**Total**: 1153

RF Commercial Services	**Precision**	**Recall**	**F1-score**	**Support**
max	0.31	0.35	0.33	314
min	0.36	0.33	0.34	295
both	0.39	0.35	0.37	258
none	0.27	0.33	0.30	284
Weighted Avg	0.33	0.34	0.33	**Total**: 1151

GBM Commercial Services	**Precision**	**Recall**	**F1-score**	**Support**
max	0.32	0.33	0.32	314
min	0.33	0.35	0.34	295
both	0.28	0.34	0.31	258
none	0.32	0.49	0.39	284
Weighted Avg	0.32	0.38	0.34	**Total**: 1151

RF Technology Services	**Precision**	**Recall**	**F1-score**	**Support**
max	0.34	0.29	0.31	355
min	0.26	0.30	0.28	352
both	0.28	0.33	0.30	102
none	0.32	0.38	0.35	337
Weighted Avg	0.30	0.32	0.31	**Total**: 1146

GBM Technology Services	**Precision**	**Recall**	**F1-score**	**Support**
max	0.34	0.30	0.32	355
min	0.30	0.31	0.30	352
both	0.27	0.35	0.30	102
none	0.29	0.37	0.33	337
Weighted Avg	0.31	0.33	0.32	**Total**: 1146

Table 6.6: Comparison of Random Forest (RF), left, and Gradient Boosting Machine (GBM), right, Classification Results for Finance, Commercial Services, and Technology Services Sectors. Precision, Recall, F1-score, and Support values are shown for each category (max, min, both, none) across both models.

In our analysis of the classification results, we first note that, despite the seemingly low classification scores, it's important to consider the challenge of predicting four distinct classes. Given that all scores are above 0.25, the models demonstrate a clear advantage over random classification, indicating some predictive power.

For the Finance sector, both models exhibit relatively similar performances across the labels, with F1-scores ranging between 0.32 and 0.36. The RF model shows slightly better recall for the 'none' class (0.35) compared to the GBM model (0.29), indicating that, in this sector, the RF model is more adept at identifying periods without turning points. However, given the similar weighted average scores for each metric $(0.33, 0.35, 0.34)$ vs $(0.37, 0.34, 0.35)$, corresponding to precision, recall and F1-score respectively, we conclude that the overall difference in performance between the two models in this sector is insignificant.

In the Commercial Services sector, the GBM model slightly outperforms the RF model when predicting periods of market stability, highlighted by the precision and recall for the 'none' class (0.32 and 0.49 against the random forest's 0.27 and 0.33). This sector demonstrates a slightly more distinct separation in performance between the two models across individual classes. The RF model performs slightly better in predicting periods with multiple turning points (with 'both' having an F1-score of 0.37 compared to GBM's 0.31), but overall, neither model has a clear advantage in this sector either.

Similarly to the Finance sector, the RF and GBM models perform closely across most classes in the Technology Services sector. For instance, for 'max' there is a negligible difference in F1-scores (0.31 for RF vs 0.32 for GBM), while the 'min' class is also comparable (0.28 for RF vs 0.30 for GBM). The models achieve identical F1-scores of 0.30 for the 'both' class, and the difference for 'none' (0.35 vs 0.33) is minor as well, again not showing a clear advantage for either model. Overall, the weighted average metrics confirm that the strengths and weaknesses of the models are consistent in Technology Services.

Taken together, the results suggest that while the Random Forest and Gradient Boosting Classifiers perform similarly across most sectors, sector-specific characteristics resulting from each sector's unique set of key momentum drivers may give one model a slight edge in predicting particular classes. Nonetheless, neither of the ensemble turning point prediction models consistently outperforms the other in any sector.

6.5 Summary

This chapter presents the results of the empirical analyses conducted for predicting momentum turning points, discussing the key features driving momentum in stock prices, and evaluating the performance of predictive models. It systematically outlines the results of feature engineering, turning point labelling, feature selection and classification, culminating in a comparison of Random Forest and Gradient Boosting models across multiple sectors.

1. Feature Engineering: The first part of the chapter revisits two approaches to feature engineering: the initial rate-of-change transformation approach, and the more successful sector-relative feature engineering method. The sector-relative approach proved more effective by providing higher-frequency data that captured company fundamentals relative to its peers on a weekly basis. This method generated features, such as 'ff_sales' and 'ff_pbk', representing deviations in a company's fundamentals from the sector average. The engineered features offered valuable insights into a company's performance and were subsequently used to predict future momentum turning points.

2. Turning Point Labels: The second section discusses the validity of the process to label momentum turning points. After dismissing the Deviation Algorithm, the Max-Min Windowing Algorithm was adopted for identifying turning points. This method effectively captured significant peaks and troughs in stock price movements within predefined windows. Label distribution analysis showed that momentum turning points (max and min) accounted for around 2% of total labels each. Visual validation of the algorithm confirmed that turning points occurred at the important shifts in stock price trend direction, supporting the reliability of the labelling process.

3. Key Momentum Drivers: After labelling turning points, the chapter moves on to identifying key momentum drivers through a multi-step feature selection process. Sector-relative features including the relative price-to-book ratio, total interest expenses, and market value, were identified as key contributors to predicting momentum. While certain features were highly valued by one method, they were excluded by another, highlighting the importance of using multiple methods to refine the selection of features. Similarly, while certain features were identified as important across all sectors, others proved highly informative for specific sectors but irrelevant in others.

4. Class Predictions: The final section of the chapter evaluates the performance of Random Forest (RF) and Gradient Boosting Machine (GBM) classifiers using the identified key momentum drivers. The results were displayed for 3 sectors of the total 19 for comparative analysis: Finance, Commercial Services, and Technology Services. Both RF and GBM models performed comparably in the Finance and Technology Services sectors, though slight differences were observed for a few turning point categories. In the Commercial Services sector, GBM slightly outperformed RF in predicting periods without turning points, demonstrating a stronger ability to capture market stability, whilst RF gained a slight edge over GBM in predicting periods with multiple turning points.

The chapter establishes that while the Random Forest and Gradient Boosting models performed similarly overall, sector-specific characteristics led to slight performance differences in predicting certain types of momentum turning points. These models serve as a reliable foundation for accurately forecasting stock momentum strength across various market sectors.

Chapter 7

Conclusions and Future Work

This chapter summarises the key findings of the thesis, evaluates them against the initial objectives, and assesses their achievement. It also highlights the strengths and limitations of the methodologies used, offering suggestions for further research and future developments in momentum trading.

7.1 Conclusions

This study developed a comprehensive framework for identifying and predicting momentum turning points in stock markets across various sectors using corporate fundamentals. Through the use of carefully engineered and selected features, along with meticulously designed labels and the utilisation of tree-based machine learning models, the research successfully demonstrated an improvement in the prediction accuracy of turning point detection compared to a benchmark random classifier. We outline how the thesis successfully met the research objectives initially set out:

1. **Define and identify momentum turning points.**

An operational definition of a series of momentum turning points was clearly established as "a sequence of significant alternating local maximums and minimums in a 10 day moving average-smoothed price series". Using historical price data, the innovative Max-Min Windowing Algorithm was developed and applied to identify significant turning points in stock momentum. This approach accurately captured local maxima and minima, aligning with theoretical definition, ensuring consistency and precision in identifying these shifts.

2. **Identify important sector-specific features.**

The study leveraged extensive historical financial data to conduct feature engineering and selection tailored to different market sectors. By performing a multi-step feature selection process, involving variance thresholding, Random Forest feature importance ranking, and correlation and information analysis, the most relevant features for each sector were identified. These 'key momentum drivers' provided sector-specific insights into stock performance, capturing critical financial metrics such as profitability, capital structure, and market valuation.

3. **Develop an appropriate methodology for predicting turning points.**

A robust predictive pipeline was developed using Random Forest (RF) and Gradient Boosting Machine (GBM) algorithms. These models were designed to anticipate stock momentum turning points across multiple sectors. The methodology balanced computational efficiency and model accuracy, employing important sector-relative features and tuning hyperparameters to optimise prediction performance across different market conditions.

4. **Evaluate model performance and ensure model robustness.**

The predictive models were rigorously evaluated using appropriate classification metrics such as precision, recall and F1-scores. To ensure reliability and consistent performance, the models were tuned using cross-validation across multiple validation sets. The results successfully demonstrated the models' effectiveness in predicting momentum turning points.

7.2 Strengths and Limitations

The study's strengths are evident throughout the construction of the pipeline. Sector-relative feature engineering provided granular insights by comparing a company's performance to its peers, resulting in more dynamic and sensitive features. The comprehensive multi-step feature selection process effectively refined the dataset by removing redundant features and retaining the most relevant ones. The Max-Min Windowing Algorithm for turning point labelling proved robust, accurately capturing significant momentum shifts with minimal false positives. Lastly, the implementation of tree-based models (Random Forest and Gradient Boosting Machine), enhanced through hyperparameter tuning and class-weight adjustments, delivered solid predictive performance, effectively addressing the challenges presented by the imbalanced label distribution.

The study also faced several limitations that warrant consideration. In feature engineering, while sector-relative comparisons were effective, they may have introduced biases if sector averages did not fully capture individual stock behaviors, potentially misrepresenting the true relative performance of certain companies. Moreover, the feature construction methodology did not account for sub-sector or market-wide dynamics, which could have provided a more comprehensive view of the factors influencing stock momentum. Additionally, the frequency of feature updates, being weekly, might have been too low to capture more granular changes in momentum. Trends in price and volume were also not incorporated into the models, which could have added valuable technical signals for momentum shifts. Lastly, while the tree-based models effectively handled the data's complexity, they pose challenges in interpretability, making it difficult to understand the precise impact of individual features on predictions. These models also struggle to capture sequential trends, since they focus on snapshot feature interactions. Addressing these limitations in future research would likely lead to more robust and accurate turning point models.

7.3 Future Work

There are several promising avenues for further research in this domain that are yet to be explored.

In this study, sector averages were used to benchmark a company's relative performance. Expanding this to include industry and market-wide averages could create a more comprehensive multi-level benchmarking system, offering a more precise reflection of a company's fundamentals. Industry averages would capture sub-sector dynamics, while market-wide averages would provide broader contexts in analysis of company performance.

Improving data quality is another potential area of focus. In this study, the presence of missing data led to the exclusion of certain features which may have contained valuable information. Future research could explore generating synthetic data or integrating data from multiple sources to fill these gaps, ensuring a more complete dataset. Incorporating secondary technical features such as price and volume trends into the models could also provide critical insights. Price trends help signal direction, while volume can validate the strength of those movements, offering additional context for predicting momentum turning points. The feature selection process could also be improved by using methods that are specifically tailored to multivariate time series data, such as Granger causality analysis. This tests whether one time series can be used to predict another, offering insights into the causal relationships between features and resulting stock price movements over time, which is more rigorous than relying solely on static importance measures.

In addition to the current model predicting four distinct turning point categories ('max', 'min', 'both', and 'none'), an alternative approach could involve developing separate binary classification models for maximums and minimums. This would simplify the task by allowing the models to focus on each type of turning point independently, increasing the explainability of our models. The performance of these binary models could be compared to the original four-class approach to assess which yields more reliable results. Additionally, regime shift detection through unsupervised learning, for example use of Hidden Markov Models (HMMs), could be explored to identify turning points by modelling transitions between bullish and bearish market regimes, offering another comparison point to evaluate model effectiveness in detecting momentum shifts.

Finally, stock momentum could be continuously ranked to provide a more robust investment signal for momentum trading strategies. This can be achieved by implementing and comparing various models, such as linear regression, tree-based methods, or recurrent neural networks, in their ability to generate momentum scores for individual stocks. A sorting algorithm could then rank these stocks based on their predicted momentum. The top-performing stocks, identified as having strong upside momentum, could be bought by investors, enabling long-only strategies to capitalise on stocks expected to experience positive price movements, thereby potentially generating profit.

7.4 Final Comments

This research provides novel insights for investors and fund managers, particularly in identifying key market shifts that support more informed decision-making for low-frequency trading strategies. By combining novel labelling techniques with innovative feature engineering and advanced modelling approaches, the study enhances our current understanding of momentum drivers and turning point prediction. The framework developed is scalable and adaptable to varying market conditions, offering a valuable tool for long-term equities investment planning.

Bibliography

[1] Christian L. Goulding, Campbell R. Harvey, and Michele Mazzoleni. Momentum turning points. 2023. Available at SSRN: `https://ssrn.com/abstract=3489539` or `http://dx.doi.org/10.2139/ssrn.3489539`.

[2] Muxi Cheng. Simple examples of cross-validation. RPubs Article, February 2023. Accessed: 2023-08-04.

[3] Maria Eslykke Søndergaard. The momentum effect on stock markets: A literature review and an empirical study. July 2010. M.Sc. Finance and Accounting (Cand.merc.FIR).

[4] Robert Novy-Marx. Fundamentally, momentum is fundamental momentum. *NBER Working Paper No. w20984*, 2015.

[5] William F. Sharpe. Capital asset prices: A theory of market equilibrium under conditions of risk. *Journal of Finance*, 19(3):425–442, September 1964.

[6] Eugene F. Fama and Kenneth R. French. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33:3–56, July 1993. Received July 1992, final version received September 1992.

[7] Taisei Kaizoji and Michiko Miyano. Stock market crash of 2008: An empirical study of the deviation of share prices from company fundamentals. *Applied Economics Letters*, 2018.

[8] Noah Beck, Shingo Goto, Jason C. Hsu, and Vitali Kalesnik. The duality of value and mean reversion. 2017.

[9] Yongling Zhang, Guihua Lu, and Wenyun Zhou. Study on differences in correlation between the accounting data and stock price of china and thailand base on ohlson model. 2018.

[10] Dashan Huang, Huacheng Zhang, and Guofu Zhou. Twin momentum: Fundamental trends matter. *SSRN*, 2019.

[11] Anwer S. Ahmed and Irfan Safdar. Dissecting stock price momentum using financial statement analysis. *Accounting and Finance*, 2018.

[12] What can explain momentum? evidence from decomposition. *Management Science*, 68(8):6184–6218, 2022.

[13] Mark Grinblatt and Bing Han. Prospect theory, mental accounting, and momentum. *Journal of Financial Economics*, 78(2):311–339, 2005.

[14] Adrian Furnham and Hua Chu Boo. A literature review of the anchoring effect. *The Journal of Socio-Economics*, 40(1):35–42, 2011.

[15] Yuxuan Huang, Luiz Capretz, and Danny Ho. Machine learning for stock prediction based on fundamental analysis. 2022.

[16] Ahmed Elbeltagi, Nikul Kumari, Jaydeo Dharpure, Ali Mokhtar, Karam Alsafadi, Manish Kumar, Behrouz Mehdinejadiani, Hadi Ramezani Etedali, Youssef Brouziyne, Abu Islam, and Alban Kuriqi. Prediction of combined terrestrial evapotranspiration index (ctei) over large river basin based on machine learning approaches. *Water*, 13(4):547, 2021.

[17] Konrad Zuchniak. Multi-teacher knowledge distillation as an effective method for compressing ensembles of neural networks. 2023. Unpublished work.

[18] Gary Antonacci. Risk premia harvesting through dual momentum. *Journal of Management & Entrepreneurship*, 2(1):27–55, 2017. Available at SSRN: https://ssrn.com/abstract=2042750 or http://dx.doi.org/10.2139/ssrn.2042750.

[19] Pedro Barroso and Pedro Santa-Clara. Momentum has its moments. *Journal of Financial Economics*, 116(1):111–120, 2015.

[20] Paul Jusselin, Edmond Lezmi, Hassan Malongo, Côme Masselin, Thierry Roncalli, and Tung-Lam Dao. Understanding the momentum risk premium: An in-depth journey through trend-following strategies. 2017. Available at SSRN: https://ssrn.com/abstract=3042173 or http://dx.doi.org/10.2139/ssrn.3042173.

[21] Thierry Roncalli. Keep up the momentum. 2017. Available at SSRN: https://ssrn.com/abstract=3083921 or http://dx.doi.org/10.2139/ssrn.3083921.

[22] Xue-Zhong 'Tony' He, Kai Li, and Youwei Li. Asset allocation with time series momentum and reversal. 2017. Available at SSRN: https://ssrn.com/abstract=2919122 or http://dx.doi.org/10.2139/ssrn.2919122.

[23] Youwei Li and Jiadong Liu. Momentum and the cross-section of stock volatility. *Journal of Economic Dynamics and Control*, 144:104524, 2022. Received 9 November 2021.

[24] Middleton Private Capital. Technology market cycles, 2023. Accessed: 2024-08-18.

Appendix A

Columns Removed and Retained During Preprocessing and Feature Selection

We provide feature removal and retainment results for the Finance sector fundamentals. The fundamentals/features sets are provided in the order they were obtained.

A.1 Non-Double-Type Columns Removed

Column Name	**Definition**
factset_entity_id	Unique identifier for a company/entity in the FactSet database.
factset_sector_desc	Description of the sector to which the company belongs according to FactSet.
fsym_id	Duplicate FactSet unique symbol identifier for the company.
adjdate	Date on which adjustments are applied to the fundamentals data.
currency	The currency in which the financial data is reported.
ff_fpnc	Identifier for the financial period type (e.g., quarterly, yearly).
ff_actg_standard	Accounting standard used by the company (e.g., IFRS, GAAP).
ff_fy_length_days	The number of days in the company's fiscal year.
ff_source_is_date	Date when the income statement was sourced or published.
ff_source_bs_date	Date when the balance sheet was sourced or published.
ff_source_cf_date	Date when the cash flow statement was sourced or published.
ff_dps_ddate	The dividend payment date (the day dividends are declared).
ff_source_doc	Source document for the reported financial data (e.g., financial report, regulatory filing).
ff_fp_ind_code	Industry code or indicator for the company according to FactSet classifications.
ff_report_freq_code	Reporting frequency code, indicating whether data is reported quarterly, annually, etc.
ff_fiscal_date	The fiscal period end date for the reported data.
ff_fyr	Fiscal year end, typically represented by the last month of the fiscal year.
ff_dps_exdate	Ex-dividend date, the date when the stock trades without the dividend.
ff_restate_ind	Indicator of whether the financial data was restated.
ff_curn_doc	The document currency in which financial data is originally reported.
ff_upd_type	The type of update applied to the financial data (e.g., new entry, revision, or restatement).

Table A.1: The 20 Removed Non-Double-Type Columns and Their Definitions

A.2 Features Dropped Due to Missing Values Thresholding

Column Name	Definition
ff_int_inc	Interest income, typically from loans or investments.
ff_non_int_inc	Non-interest income, such as fees or service charges.
ff_gross_inc	Total gross income of the company.
ff_loan_loss_prov	Provision for loan losses, reflecting potential defaults.
ff_non_int_exp	Non-interest expenses, such as operational costs.
ff_cash_st	Cash and short-term investments available.
ff_cash_only	Cash available for immediate use.
ff_cash_due_fr_bk	Cash due from other banks or financial institutions.
ff_receiv_tot	Total receivables, including money owed to the company.
ff_inven	Inventory held by the company.
ff_assets_curr	Current assets, expected to be liquidated or used within one year.
ff_deps	Deposits held by the company, typically in financial institutions.
ff_liabs_curr	Current liabilities, expected to be settled within one year.
ff_pay_acct	Accounts payable, or obligations to pay off short-term debts.
ff_accr_exp_xpayr	Accrued expenses excluding payroll, expenses that have been incurred but not yet paid.
ff_ins_liabs_pol	Insurance liabilities related to policy obligations.
ff_dfd_tax_itc	Deferred tax and investment tax credits, liabilities or assets to be settled in future periods.
ff_liabs_xmin_int_accum	Liabilities excluding minority interests and accumulated items.
ff_tier1_cap	Tier 1 capital, a core measure of a bank's financial strength.
ff_tier2_cap	Tier 2 capital, supplementary capital that includes items like revaluation reserves.
ff_net_inc_cf	Net income derived from cash flows.
ff_dep_exp_cf	Depreciation expense as part of cash flow adjustments.
ff_non_cash	Non-cash items that affect the financial statements but not cash flow.
ff_loan_incr_cf	Cash outflows due to the increase in loans issued.
ff_loan_decr_cf	Cash inflows due to the decrease in loans issued.
ff_deps_decr_cf	Decrease in deposits as reflected in cash flows.
ff_deps_incr_cf	Increase in deposits as reflected in cash flows.
ff_for_exch_cf	Foreign exchange-related cash flow activities.
ff_pay_tax	Taxes paid by the company.
ff_dfd_tax	Deferred tax liabilities or assets.
ff_loan_net	Net loan amounts after accounting for issuance and repayments.
ff_deps_cust	Customer deposits, typically held by banks and financial institutions.
ff_receiv_st	Short-term receivables, expected to be collected within one year.
ff_dfd_tax_db	Deferred tax debits, representing future tax benefits.
ff_dfd_tax_cr	Deferred tax credits, representing future tax liabilities.
ff_receiv_int	Interest receivable, the amount of interest income yet to be received.
ff_cash_restr	Restricted cash, not available for general use due to contractual or legal restrictions.
ff_cust_accept	Customer acceptance liabilities, often seen in financial institutions dealing with trade finance.
ff_ebit_oper_ps	Earnings before interest and taxes (EBIT) per share from operating activities.
ff_invest_re	Real estate investments held by the company.
ff_notes_receiv_lt	Long-term notes receivable, to be collected after one year.

ff_prem_receiv	Premiums receivable, typically for insurance companies, representing amounts due for policies issued.
ff_div_pay_out_ps	Dividends paid out per share.
ff_invest_yld_5yavg	Five-year average yield on investments, measuring the returns from invested assets.
ff_secs_custody	Securities held in custody, typically by banks or brokers on behalf of clients.
ff_inven_lifo	Inventory valued using the Last-In, First-Out (LIFO) accounting method.
ff_rent_inc	Rental income, generated from leasing out properties or equipment.
ff_bk_com_eq_tier1_tot	Total Tier 1 capital as a proportion of the bank's common equity.
ff_cap_lease_curr	Current portion of capital lease obligations due within one year.
ff_curr_ins_ben	Current insurance benefits payable, liabilities for benefits owed to policyholders.
ff_dfd_tax_cf	Deferred tax adjustments affecting the company's cash flows.
ff_oper_lease_repay	Repayments related to operating leases.
ff_bnfit_loss_rsrv_tcap	Benefit loss reserves related to total capital, often seen in insurance companies.
ff_capex_fix_assets	Capital expenditures on fixed assets, such as property, plant, and equipment.
ff_cash_curr_assets	Cash as a proportion of current assets.
ff_cash_secs_deps	Cash, securities, and deposits, indicating liquid assets.
ff_cogs_sales	Cost of goods sold as a proportion of sales, measuring efficiency in producing goods.
ff_com_eq_deps	Common equity as a proportion of total deposits, common in financial institutions.
ff_dep_accum_fix_assets	Accumulated depreciation on fixed assets such as buildings and machinery.
ff_deps_assets	Deposits as a proportion of total assets, showing the relationship between deposits and the company's asset base.
ff_dfd_tax_assets_lt	Long-term deferred tax assets, representing future tax benefits.
ff_earn_assets_pct	Percentage of earning assets (assets generating income) in the company's portfolio.
ff_ebit_oper_mgn	Operating margin calculated using EBIT (Earnings Before Interest and Taxes), showing profitability from operations.
ff_ebit_oper_roa	Return on assets using EBIT, measuring how effectively the company uses its assets to generate profits from operations.
ff_ebitda_cf	EBITDA (Earnings Before Interest, Taxes, Depreciation, and Amortization) derived from cash flow.
ff_ebitda_oper_mgn	Operating margin calculated using EBITDA, indicating operational efficiency.
ff_entrpr_val_ebit_oper	Enterprise value as a multiple of EBIT from operating activities, a valuation metric.
ff_entrpr_val_ebitda_oper	Enterprise value as a multiple of EBITDA from operating activities, commonly used in valuation.
ff_ebit_oper_fix_chrg_covg	Fixed charge coverage ratio based on EBIT, measuring the company's ability to cover fixed expenses.
ff_ins_rsrv_gr	Insurance reserves growth, common in insurance companies, reflecting changes in reserves held.
ff_ebit_oper_int_covg	Interest coverage ratio using EBIT, indicating the company's ability to cover interest payments.
ff_int_exp_ib_liabs	Interest expenses on interest-bearing liabilities, showing costs related to borrowed funds.
ff_int_inc_avg_deps	Interest income as a proportion of average deposits.
ff_int_inc_earn_assets	Interest income from earning assets, typically loans and investments.
ff_inven_curr_assets	Inventory as a proportion of current assets.
ff_inven_days	Days inventory outstanding, indicating how long inventory is held before being sold.
ff_inven_turn	Inventory turnover ratio, measuring how efficiently inventory is managed.

ff_invest_assets_deps	Investment assets as a proportion of total deposits.
ff_invest_assets_loan_deps	Investment assets and loans as a proportion of total deposits, indicating a financial institution's investment strategy.
ff_invest_inc_invest_assets	Investment income from investment assets, showing returns from investments.
ff_invest_yld	Yield on investments, measuring the returns generated by investment assets.
ff_loan_gr	Loan growth, indicating the increase or decrease in loan issuance over a period.
ff_loan_loss_rsrv_assets	Loan loss reserves as a proportion of total assets, reflecting potential losses on loans.
ff_loan_loss_rsrv_tcap	Loan loss reserves as a proportion of total capital, indicating the company's ability to absorb loan defaults.
ff_loss_ratio	Loss ratio, commonly used in insurance, showing the proportion of claims paid to premiums earned.
ff_net_int_inc_earn_assets	Net interest income generated from earning assets like loans and investments.
ff_non_int_inc_rev	Non-interest income as a proportion of total revenue.
ff_oper_inc_prem_earn	Operating income derived from premiums earned, typically for insurance companies.
ff_pay_acct_sales	Accounts payable as a proportion of sales, indicating short-term obligations relative to revenue.
ff_ppe_net_soft_equip	Net property, plant, and equipment, including software and equipment, after depreciation.
ff_rd_sales	Research and development expenses as a proportion of sales, indicating the company's investment in innovation.
ff_receiv_curr_assets	Receivables as a proportion of current assets.
ff_receiv_turn	Receivables turnover ratio, measuring how efficiently a company collects its receivables.
ff_receiv_turn_days	Days receivables outstanding, showing the average time to collect receivables.
ff_roea	Return on earning assets, indicating profitability from assets that generate income.
ff_sales_fix_assets	Sales as a proportion of fixed assets, showing asset efficiency in generating revenue.
ff_sales_inven_turn	Sales as a function of inventory turnover, measuring the relationship between sales and inventory management.
ff_sales_wkcap	Sales as a proportion of working capital, indicating how efficiently a company uses its working capital to generate revenue.
ff_sga_oth	Selling, general, and administrative (SG&A) expenses excluding other specific items.
ff_sga_sales	SG&A expenses as a proportion of sales, indicating the cost structure relative to revenue.
ff_spec_items	Special items, often non-recurring or one-time expenses or gains.
ff_tcap_deps	Total capital as a proportion of deposits, showing the capital strength relative to deposits.
ff_unearn_prem_tcap	Unearned premiums as a proportion of total capital, relevant to insurance companies, showing liabilities for premiums not yet earned.
ff_wkcap	Working capital, the difference between current assets and current liabilities.
ff_wkcap_pct	Working capital as a percentage of total assets, showing liquidity.
ff_zscore	Z-score, a statistical measure typically used for bankruptcy prediction or financial distress.
ff_debt_serv	Debt service, representing the amount required to cover the repayment of interest and principal on a debt.
ff_pfcf	Price to free cash flow ratio, indicating how much investors are willing to pay for a dollar of free cash flow.
ff_pfcf_dil	Price to diluted free cash flow ratio, considering potential dilution from securities like convertible debt.

ff_receiv_gross	Gross receivables, representing the total amount owed to the company before any allowances for doubtful accounts.
ff_receiv_st_oth	Other short-term receivables, representing money owed to the company in the short term, excluding trade receivables.
ff_loan_assets	Loans as a proportion of total assets, showing the extent of loans in the company's asset portfolio.
ff_loan_deps	Loans as a proportion of deposits, indicating a bank's loan-to-deposit ratio.
ff_loan_gross	Gross loans issued by the company before deducting reserves for losses.
ff_loan_loss_prov_pct	Provision for loan losses as a percentage of total loans, indicating expected defaults.
ff_loan_loss_rsrv_pct	Loan loss reserves as a percentage of total loans, representing the buffer for potential loan defaults.
ff_loan_tcap	Loans as a proportion of total capital, showing the relationship between loans and capital.
ff_nonperf_loan_pct	Non-performing loans as a percentage of total loans, indicating the portion of loans that are in default or not generating income.
ff_pay_turn_days	Days payable outstanding, representing the average number of days the company takes to pay its suppliers.
ff_int_inc_misc	Miscellaneous interest income, representing other sources of interest revenue.
ff_fin_invest_tot	Total financial investments, including securities and other financial instruments held.
ff_trade_inc_net	Net trade income, often representing profits from trading activities in financial institutions.
ff_bk_oper_exp_oth	Other bank operating expenses, representing additional operating costs in banking activities.
ff_bk_oper_exp_tot	Total bank operating expenses, capturing all operating costs for financial institutions.
ff_invest_st_tot	Total short-term investments, including liquid securities and other short-term financial instruments.
ff_nonperf_loan_com_eq	Non-performing loans as a proportion of common equity, indicating the risk posed to equity from loan defaults.
ff_nonperf_loan_loss_rsrv	Non-performing loans covered by loan loss reserves, indicating how much of the non-performing loans are covered by reserves.
ff_fscore	F-score, a financial score indicating the company's overall financial health and performance.
ff_impair	Impairment charges, representing write-downs in the value of assets.
ff_oth_xcept_chrg	Other exceptional charges, representing one-time or non-recurring costs.
ff_restruct_debt	Debt restructuring, representing adjustments or reorganizations of the company's debt.
ff_compr_inc	Comprehensive income, including net income and other comprehensive items such as unrealized gains or losses.
ff_compr_inc_for_curn_adj	Adjustments to comprehensive income for currency translation, reflecting gains or losses from foreign currency fluctuations.
ff_compr_inc_oth	Other comprehensive income, capturing additional non-operating gains or losses.
ff_compr_inc_pens_liabs_adj	Adjustments to comprehensive income related to pension liabilities.
ff_compr_inc_tot	Total comprehensive income, including net income and other comprehensive income.
ff_compr_inc_unreal_gl_secs	Unrealized gains or losses on securities, affecting comprehensive income.
ff_int_fin_cf	Interest paid from financial activities in the cash flow statement.
ff_int_oper_cf	Interest paid from operating activities in the cash flow statement.
ff_liabs_lease_lt	Long-term lease liabilities, representing obligations beyond one year.
ff_liabs_lease_st	Short-term lease liabilities, representing obligations within one year.

ff_accr_exp_cf	Accrued expenses included in the cash flow statement.
ff_amort_cf	Amortisation expenses reported in the cash flow statement.
ff_assets_sep_accts	Assets held in separate accounts, typically for specific purposes like insurance or retirement funds.
ff_com_eq_for_exch	Foreign exchange adjustments to common equity.
ff_com_eq_hedg_gl	Gains or losses from hedging activities affecting common equity.
ff_com_eq_oth_compr_adj_oth	Other comprehensive adjustments to common equity, not classified elsewhere.
ff_com_eq_unearn_comp	Unearned compensation impacting common equity, such as stock options or deferred payments.
ff_commiss_inc	Commission income earned from providing services, such as brokerage fees.
ff_dep_exp_uncon	Depreciation expense for unconsolidated entities.
ff_dep_exp_xamort_cf	Depreciation expenses excluding amortisation, reported in the cash flow statement.
ff_dfd_tax_xitc_cf	Deferred tax adjustments and investment tax credits reflected in the cash flow statement.
ff_div_pfd_cf	Dividends paid on preferred stock, reported in the cash flow statement.
ff_eps_headline_uk	Headline earnings per share, a common metric used in the UK for corporate profitability.
ff_eps_uncon	Earnings per share for unconsolidated entities.
ff_equip_exp	Equipment expenses, often related to purchasing or maintaining company equipment.
ff_fed_funds	Federal funds, typically referring to the funds lent between banks in the Federal Reserve system.
ff_for_curn_adj	Adjustments for foreign currency fluctuations, affecting financial statements.
ff_for_exch_bs	Foreign exchange gains or losses reported on the balance sheet.
ff_for_exch_is	Foreign exchange gains or losses reported on the income statement.
ff_ins_invest_inc	Investment income generated from insurance-related investments.
ff_ins_liabs_oth	Other insurance-related liabilities, such as claims or policy obligations.
ff_ins_lt_rsrv	Long-term insurance reserves, typically for claims or policyholder obligations.
ff_ins_rsrv	Insurance reserves, representing liabilities for future claims.
ff_int_exp_deps	Interest expenses related to customer deposits, typically for financial institutions.
ff_int_exp_fed_repos	Interest expenses on federal repos (repurchase agreements), common in banking.
ff_int_exp_oth_borr	Interest expenses on other borrowings, excluding deposits or federal funds.
ff_int_inc_deps	Interest income generated from customer deposits.
ff_int_inc_fed_funds	Interest income from federal funds, often seen in interbank lending.
ff_int_inc_loan	Interest income generated from loans issued by the company.
ff_int_inc_oth	Other interest income from miscellaneous sources.
ff_intang_oth_amort	Amortization of other intangible assets, such as patents or trademarks.
ff_intang_oth_gross	Gross value of other intangible assets before amortization.
ff_inven_cf	Changes in inventory levels as reported in the cash flow statement.
ff_invest_inc	Income generated from investments, such as stocks, bonds, or real estate.
ff_labor_exp	Labor expenses, including wages and employee benefits.
ff_loan_bk	Loans held by the bank, representing assets in the form of issued loans.
ff_min_pens_liabs_adj	Adjustments to pension liabilities for minimum funding requirements.
ff_pay_acct_cf	Changes in accounts payable as reported in the cash flow statement.
ff_pay_tax_cf	Taxes paid as reported in the cash flow statement.
ff_pay_tax_dfd_tax	Deferred tax liabilities or assets related to taxes payable.
ff_pol_claims	Policyholder claims, relevant to insurance companies, representing liabilities for claims.
ff_ppe_dep_soft_equip	Depreciation on property, plant, and equipment, including software and equipment.

ff_ppe_gross_constr	Gross value of property, plant, and equipment related to construction.
ff_ppe_gross_soft_equip	Gross value of software and equipment before depreciation.
ff_prem_unearn	Unearned premiums, representing insurance premiums received but not yet recognized as revenue.
ff_real_gain	Gains from real estate transactions or investments.
ff_receiv_cf	Changes in receivables as reported in the cash flow statement.
ff_rsrv_appr_oth	Other reserves or appropriations not classified elsewhere.
ff_sales_uncon	Sales from unconsolidated entities, representing revenue from partially owned businesses.
ff_secs_gain	Gains from the sale of securities, contributing to investment income.
ff_tax_non_inc	Non-income taxes paid, including property, sales, or other non-income related taxes.
ff_trade_acct	Trading account, representing financial assets held for trading purposes.
ff_trade_inc	Trading income, typically from financial institutions, representing profits from trading activities.
ff_trust_inc	Trust income, earned from fiduciary or custodial services.
ff_wkcap_assets_oth	Other working capital assets, not classified as cash, receivables, or inventory.
ff_assets_risk_wght	Risk-weighted assets, used in determining regulatory capital requirements for financial institutions.
ff_compr_inc_pens_liabs	Comprehensive income adjustments related to pension liabilities.
ff_fix_assets_impair	Impairment charges on fixed assets, indicating a reduction in their carrying value.
ff_ppe_impair	Impairment charges on property, plant, and equipment.
ff_tax_cf	Taxes paid as reported in the cash flow statement.
ff_cap_ratio_tier1	Tier 1 capital ratio, a key regulatory measure of a bank's financial strength.
ff_cap_ratio_tot	Total capital ratio, including Tier 1 and Tier 2 capital, used to assess a bank's solvency.
ff_comp_soft	Computer software, often capitalized as an intangible asset.
ff_int_cf	Interest paid or received as reported in the cash flow statement.
ff_bdebt	Bad debt expenses, representing uncollectible amounts from customers or borrowers.
ff_impair_ppe	Impairment charges on property, plant, and equipment (PPE).
ff_fin_assets_impair	Impairment charges on financial assets, such as investments or receivables.
ff_reorg_restruct_exp	Reorganization and restructuring expenses, including costs related to layoffs or changes in business structure.
ff_gw_wdown	Goodwill write-down, representing the reduction in the carrying value of goodwill.
ff_legal_claim_exp	Expenses related to legal claims or litigation.
ff_impair_intang_oth	Impairment charges on other intangible assets.
ff_unreal_invest_gl	Unrealized gains or losses on investments, affecting comprehensive income but not yet realized.
ff_oth_unusual_exp	Other unusual expenses, typically non-recurring or one-time charges.
ff_deps_demand	Demand deposits, representing funds that can be withdrawn by depositors at any time without notice.
ff_receiv_cf	Changes in receivables reported in the cash flow statement.
ff_rsrv_appr_oth	Other reserves or appropriations not classified elsewhere.
ff_sales_uncon	Sales from unconsolidated entities, revenue from partially owned businesses.
ff_secs_gain	Gains from the sale of securities, contributing to investment income.
ff_tax_non_inc	Non-income taxes, such as property or sales taxes.
ff_trade_acct	Trading account, representing assets held for trading purposes.
ff_trade_inc	Trading income, typically from financial activities.
ff_trust_inc	Income earned from trust and fiduciary services.
ff_wkcap_assets_oth	Other working capital assets not classified as cash, receivables, or inventory.

ff_assets_risk_wght	Risk-weighted assets, used in banking to determine regulatory capital requirements.
ff_compr_inc_pens_liabs	Comprehensive income adjustments related to pension liabilities.
ff_fix_assets_impair	Impairment charges on fixed assets, indicating a reduction in value.
ff_ppe_impair	Impairment charges on property, plant, and equipment.
ff_tax_cf	Taxes paid as reported in the cash flow statement.
ff_cap_ratio_tier1	Tier 1 capital ratio, assessing a bank's financial strength.
ff_cap_ratio_tot	Total capital ratio, including both Tier 1 and Tier 2 capital.
ff_comp_soft	Computer software, capitalized as an intangible asset.
ff_int_cf	Interest paid or received, as reported in the cash flow statement.
ff_bdebt	Bad debt expense, representing uncollectible amounts from customers or loans.
ff_impair_ppe	Impairment charges on property, plant, and equipment.
ff_fin_assets_impair	Impairment charges on financial assets like investments or receivables.
ff_reorg_restruct_exp	Reorganization and restructuring expenses.
ff_gw_wdown	Goodwill write-down, indicating a reduction in goodwill value.
ff_legal_claim_exp	Expenses related to legal claims or litigation.
ff_impair_intang_oth	Impairment charges on other intangible assets.
ff_unreal_invest_gl	Unrealized gains or losses on investments.
ff_oth_unusual_exp	Other unusual expenses, often one-time or non-recurring.
ff_deps_demand	Demand deposits, which can be withdrawn at any time without notice.
ff_loan_oth	Other loans issued by the company.
ff_prep_exp	Prepaid expenses, payments made for future services or goods.
ff_assets_curr_misc	Miscellaneous current assets not classified elsewhere.
ff_ppe_gross	Gross property, plant, and equipment value before depreciation.
ff_ppe_dep	Depreciation on property, plant, and equipment.
ff_liabs_curr_misc	Miscellaneous current liabilities not classified elsewhere.
ff_loan_rsrv	Loan reserves, funds set aside for potential loan losses.
ff_cogs_xdep	Cost of goods sold excluding depreciation.
ff_rd_exp	Research and development expenses.
ff_loan_nonperf	Non-performing loans, loans in default or not generating income.
ff_ordinary_inc	Ordinary income from core business activities.
ff_loss_claim_exp	Loss and claim expenses, typically for insurance companies.
ff_prem_earn	Earned premiums, income from insurance policies.
ff_underwriting_exp	Underwriting expenses, costs related to issuing insurance policies.
ff_debt_oth_lt_curr	Other long-term debt due within the current period.
ff_receiv_net	Net receivables, after allowances for doubtful accounts.
ff_stk_opt_cf	Stock options as reported in the cash flow statement.
ff_stk_opt_exp	Expenses related to stock options granted to employees.
ff_dfd_inc	Deferred income, revenue received but not yet earned.
ff_oper_lease_exp	Operating lease expenses.
ff_shs_repurch_total_val	Total value of shares repurchased by the company.
ff_shs_repurch_total_shs	Total number of shares repurchased by the company.
ff_amort_intang	Amortization of intangible assets.
ff_dep_exp	Depreciation expense.
ff_inven_fg	Finished goods inventory.
ff_inven_matl	Raw materials inventory.
ff_inven_prog_paymt	Payments made for inventory in progress.
ff_inven_wip	Work in progress inventory.
ff_pens_bnfit_retir_post	Post-retirement pension benefits.
ff_int_inc_fed_repos	Interest income from federal repos.
ff_unreal_gl_tot	Total unrealized gains or losses.
ff_accr_payr	Accrued payroll expenses.

ff_deps_for	Foreign deposits, held in accounts overseas.
ff_deps_sav	Savings deposits.
ff_deps_unspec	Unspecified deposits, not classified by type.
ff_dfd_chrg	Deferred charges, prepaid expenses or costs that will be recognized later.
ff_inc_unearn	Unearned income, revenue received but not yet earned.
ff_int_inc_non_oper	Interest income from non-operating activities.
ff_liabs_curr_dfd_tax	Current deferred tax liabilities.
ff_loan_brkr	Broker loans, typically short-term loans to brokers.
ff_loan_comml_indl	Commercial and industrial loans issued by the company.
ff_loan_cons	Consumer loans issued by the company.
ff_loan_for	Foreign loans, issued in foreign currencies or overseas.
ff_loan_mtge	Mortgage loans issued by the company.
ff_net_cap_require	Net capital requirements, typically for financial institutions.
ff_rsrv_unappr	Unapproved reserves.
ff_wkcap_ps	Working capital per share.
ff_xcept_prov	Exceptional provisions, one-time charges or allowances.
ff_adv_exp	Advertising expenses.
ff_dfd_inc_curr	Current deferred income.
ff_sale_ppe_cf	Proceeds from the sale of property, plant, and equipment as reported in the cash flow statement.
ff_tax_chg_eff	Effective tax charge, the company's tax expense as a percentage of pretax income.
ff_acq_process_rd	Research and development expenses related to acquisition processes.
ff_ga_exp	General and administrative expenses.
ff_sell_exp	Selling expenses.
ff_mkt_exp	Marketing expenses.
ff_liabs_oper_lease	Operating lease liabilities.
ff_liabs_oper_lease_curr	Current operating lease liabilities.
ff_assets_lease_net	Net lease assets, after depreciation.
ff_amort_exp_lease	Amortization expenses related to lease assets.
ff_int_exp_lease	Interest expenses related to lease liabilities.
ff_frank_bal_cf	Changes in the franked balance (tax-adjusted retained earnings) in the cash flow statement.
ff_debt_oth_cf	Other debt-related cash flows not classified elsewhere.
ff_accel_dep	Accelerated depreciation expenses.
ff_calamitous_event	Expenses related to calamitous or catastrophic events.
ff_early_term_contract	Costs or penalties associated with early termination of contracts.
ff_eps_headline_uk_dil	Headline earnings per share (diluted), commonly used in the UK.
ff_unreal_gl_prop	Unrealized gains or losses on properties.
ff_unreal_gl_bio_assets	Unrealized gains or losses on biological assets.
ff_unreal_gl_deriv	Unrealized gains or losses on derivatives.
ff_unreal_gl_invest	Unrealized gains or losses on investments.
ff_unreal_gl_oth	Unrealized gains or losses on other financial assets.
ff_emp_num	Number of employees.
ff_oper_prov	Operating provisions, expenses set aside for expected liabilities.
ff_pay_div	Dividends paid.
ff_cogs	Cost of goods sold.
ff_unusual_exp	Unusual expenses, typically one-time or non-recurring.
ff_int_inc_aft_prov	Interest income after provisions.
ff_loss_claim_rsrv	Loss claim reserves, typically for insurance companies.
ff_oper_inc_bef_int	Operating income before interest expenses.
ff_oper_inc_aft_int	Operating income after interest expenses.
ff_assets_curr_oth	Other current assets not classified elsewhere.

ff_curr_ratio	Current ratio, a liquidity measure comparing current assets to current liabilities.
ff_ebitda_oper	Earnings before interest, taxes, depreciation, and amortization from operations.
ff_gross_mgn	Gross margin, the ratio of gross profit to revenue.
ff_quick_ratio	Quick ratio, a stricter liquidity measure than the current ratio.
ff_int_inc_net	Net interest income, interest income minus interest expenses.
ff_sga	Selling, general, and administrative expenses.
ff_liabs_curr_oth	Other current liabilities not classified elsewhere.
ff_ebit_oper	Earnings before interest and taxes (EBIT) from operations.
ff_debt_ebitda_oper	Debt to EBITDA ratio from operations, measuring debt burden relative to earnings.
ff_int_mgn	Interest margin, the difference between interest income and interest expense as a percentage of assets.
ff_turn_rate	Turnover rate, measuring the efficiency of asset use to generate revenue.
ff_loss_adj_exp	Loss adjustment expenses, typically for insurance companies.

Table A.2: The 309 Features Dropped from the Finance Sector data due to Missing Values Thresholding

A.3 Features Dropped Due to Low Variance

Column Name	**Definition**
ff_fpnc	Financial period code, indicating the type of reporting period.
ff_eq_aff_inc	Equity-affiliated income, representing income from investments in affiliates.
ff_xord_cf	Extraordinary items in cash flow, non-recurring cash flow items.
ff_actg_standard	Accounting standard used by the company, such as IFRS or GAAP.
ff_fy_length_days	Length of the fiscal year in days.
ff_prov_risk	Provision for risk, set aside for potential future liabilities.
ff_ppe_net	Net property, plant, and equipment after depreciation.
ff_rsrv_noneq	Non-equity reserves, representing liabilities that are not tied to equity.
ff_report_freq_code	Reporting frequency code, indicating how often financial reports are issued (quarterly, annually, etc.).
ff_fyr	Fiscal year end, representing the month when the fiscal year ends.
ff_rsrv_chg	Change in reserves, typically adjustments to reserves set aside for liabilities.
ff_loan_chg_cf	Change in loans as reported in the cash flow statement.
ff_dep_chg_cf	Change in deposits as reported in the cash flow statement.
ff_fix_assets_com_eq	Fixed assets as a proportion of common equity.
ff_assets_disc_oper	Discontinued operations assets, representing assets held for sale or disposal.
ff_liabs_disc_oper	Discontinued operations liabilities, representing liabilities associated with discontinued operations.
ff_liabs_lease	Lease liabilities, representing obligations under leasing arrangements.
ff_cap_lease	Capital lease obligations, long-term leases where the lessee assumes some ownership.
ff_gw	Goodwill, an intangible asset representing the excess of purchase price over the fair value of acquired net assets.
ff_par_ps	Par value per share, the nominal value of a share of stock.
ff_pfd_stk_nred	Preferred stock not redeemable, representing a class of stock with a fixed dividend and no redemption option.
ff_treas_shs	Treasury shares, shares that the company has repurchased.

ff_treas_stk	Treasury stock, stock that has been repurchased by the company and held in its own treasury.
ff_gw_impair	Goodwill impairment, indicating a reduction in the value of goodwill.
ff_intang_oth_impair	Impairment of other intangible assets, such as trademarks or patents.
ff_actg_chg	Accounting changes, adjustments due to changes in accounting policies.
ff_capex_oth	Other capital expenditures, not directly related to property, plant, or equipment.
ff_int_cap	Interest capitalized, representing interest added to the cost of an asset rather than expensed.
ff_ppe_net_owned	Net property, plant, and equipment owned by the company after depreciation.
ff_dfd_tax_rsrv	Deferred tax reserves, representing taxes that are payable in future periods.
ff_restruct_exp	Restructuring expenses, costs associated with reorganizing the company.
ff_misc_funds_cf	Miscellaneous funds reported in the cash flow statement.
ff_fp_ind_code	Financial period industry code, used to categorize the industry for reporting purposes.

Table A.3: Features Dropped from Finance Sector data due to Low Variance in Selected Stock Sample

A.4 Top Quintile of Features from Random Forest Selection

Column Name	**Definition**
ff_pbk	Price to book ratio, indicating the market price relative to the company's book value.
ff_pbk_secs	Price to book ratio for securities.
ff_int_exp_tot	Total interest expenses incurred by the company.
ff_sales	Total sales or revenue generated by the company.
ff_ebitda_bef_unusual	EBITDA before unusual items, reflecting operating profitability.
ff_pbk_tang	Price to tangible book value ratio, excluding intangible assets from the book value.
ff_price_close_fp	Closing price at the end of the financial period.
ff_sales_gr	Sales growth year over year, measuring the annual increase in revenue.
ff_capex_assets	Capital expenditures as a percentage of total assets.
ff_com_eq	Common equity, representing the ownership stake of common shareholders.
ff_ebit_bef_unusual	Earnings before interest and taxes (EBIT), excluding unusual items.
ff_mkt_val_secs	Market value of securities held by the company.
ff_oper_exp_tot	Total operating expenses, representing the costs incurred during operations.
ff_std_debt	Standard debt, usually referring to long-term or regular debt obligations.
ff_bk_oper_inc_oth	Other banking operating income, typically from non-interest sources.
ff_com_shs_out_eps_dil	Diluted earnings per share based on outstanding common shares.
ff_earn_yld	Earnings yield, the inverse of the price-to-earnings ratio, showing earnings as a percentage of price.
ff_mkt_val	Market value, representing the total market capitalization of the company.
ff_entrpr_val	Enterprise value, representing the total value of the company, including debt and equity.
ff_commiss_inc_net	Net commission income, typically for financial institutions, derived from brokerage or advisory fees.
ff_ptx_inc	Pre-tax income, earnings before tax expenses are deducted.

ff_oper_exp_oth	Other operating expenses, covering costs not included in the main operating categories.
ff_pfd_stk	Preferred stock, representing ownership with a fixed dividend but no voting rights.
ff_fp_ind_code	Financial period industry code, used for industry classification in financial reporting.
ff_pe_secs	Price-to-earnings ratio for securities, reflecting the valuation of earnings relative to price.
ff_invest_aff	Investments in affiliates, representing equity stakes in other entities.
ff_bps_tang	Tangible book value per share, excluding intangible assets from book value.
ff_bps	Book value per share, representing equity available to shareholders on a per-share basis.
ff_net_inc_aft_xord	Net income after extraordinary items have been accounted for.
ff_pfd_stk_tcap	Preferred stock as a percentage of total capital.
ff_rotc	Return on total capital, a profitability measure relative to total invested capital.
ff_bps_secs	Book value per share for securities.
ff_psales_dil	Price to sales ratio, calculated using diluted shares outstanding.
ff_ptx_xord_chrg	Pre-tax charges for extraordinary items.
ff_com_eq_retain_earn	Retained earnings as part of common equity.
ff_pe	Price-to-earnings ratio, reflecting the valuation of earnings relative to price.
ff_int_exp_oth	Other interest expenses, excluding standard debt.
ff_com_eq_apic	Additional paid-in capital (APIC), representing the excess amount paid over the par value of stock.
ff_compr_inc_accum	Accumulated comprehensive income, including gains or losses not included in net income.
ff_int_exp_debt	Interest expense on debt obligations.

Table A.4: Top Quintile of Features given Current Stock Sample from the Finance Sector Obtained Through Random Forest Feature Importance. 1mth forward returns target variable

A.5 Dropped Features from Correlation Analysis

Column Name	**Definition**
ff_bps_secs	Book value per share for securities, representing the equity available to shareholders for each security.
ff_price_close_fp	Closing price at the end of the financial period, reflecting the stock's market value at that time.
ff_mkt_val	Market value, representing the total market capitalization of the company based on stock price and outstanding shares.
ff_bps	Book value per share, representing the equity available to shareholders on a per-share basis.
ff_com_shs_out_eps	Earnings per share based on outstanding common shares, typically used to measure profitability.
ff_pbk_secs	Price-to-book ratio for securities, comparing the market value of securities to their book value.
ff_ebit_bef_unusual	Earnings before interest and taxes (EBIT), excluding unusual or non-recurring items, representing core profitability.
ff_mkt_val_secs	Market value of securities held by the company, representing their current market worth.

Table A.5: Features Dropped from Correlation Analysis given Current Stock Sample from the Finance Sector

A.6 Dropped Features from Mutual Information Analysis

Column Name	Definition
ff_sales	Total sales or revenue generated by the company.
ff_invest_aff	Investments in affiliates, representing equity stakes in other entities.
ff_com_eq_retain_earn	Retained earnings as part of common equity, representing profits reinvested in the company rather than distributed as dividends.
ff_oper_exp_oth	Other operating expenses, including costs not classified in main operating categories.
ff_capex_assets	Capital expenditures as a percentage of total assets, indicating investment in long-term assets.
ff_std_debt	Standard debt, usually referring to long-term or regular debt obligations of the company.
ff_compr_inc_accum	Accumulated comprehensive income, including gains or losses not included in net income, such as foreign currency adjustments or unrealized gains/losses.

Table A.6: Features That Share No Mutual Information with 1 Month Forward Returns Dropped given Current Stock Sample from the Finance Sector

A.7 Remaining Features - The Key Momentum Drivers

Column Name	Definition
ff_pbk	Price to book ratio, comparing the company's market value to its book value.
ff_int_exp_tot	Total interest expenses incurred by the company.
ff_ebitda_bef_unusual	EBITDA (Earnings Before Interest, Taxes, Depreciation, and Amortization) before unusual items, reflecting core profitability.
ff_pbk_tang	Price to tangible book value ratio, excluding intangible assets.
ff_sales_gr	Year-over-year sales growth, measuring the percentage increase in sales over the previous year.
ff_com_eq	Common equity, representing the ownership stake of common shareholders in the company.
ff_price_close_fp	Closing price of the stock at the end of the financial period.
ff_oper_exp_tot	Total operating expenses, covering all costs related to company operations.
ff_bk_oper_inc_oth	Other operating income from banking activities, typically excluding interest income.
ff_com_shs_out_eps_dil	Diluted earnings per share based on the number of outstanding common shares.
ff_earn_yld	Earnings yield, calculated as earnings per share divided by the stock price, reflecting the return on investment.
ff_mkt_val	Market value, representing the total market capitalization of the company.
ff_entrpr_val	Enterprise value, representing the total value of the company, including equity and debt.
ff_commiss_inc_net	Net commission income, often generated by financial institutions from brokerage or advisory services.
ff_ptx_inc	Pre-tax income, representing earnings before taxes are deducted.
ff_pfd_stk	Preferred stock, providing a fixed dividend and priority over common stock in dividend payments.
ff_mkt_val_secs	Market value of securities, representing the value of investments in securities.
ff_fp_ind_code	Financial period industry code, used to classify the company's industry for reporting purposes.

ff_pe_secs	Price-to-earnings ratio for securities, measuring the market price relative to earnings.
ff_bps_tang	Tangible book value per share, excluding intangible assets from book value.
ff_net_inc_aft_xord	Net income after extraordinary items, representing the company's profitability after accounting for non-recurring events.
ff_pfd_stk_tcap	Preferred stock as a percentage of total capital, indicating the share of preferred equity in the company's capital structure.
ff_rotc	Return on total capital, measuring the profitability of the company's capital investments.
ff_psales_dil	Price to sales ratio based on diluted shares outstanding, measuring the stock price relative to sales per share.
ff_ptx_xord_chrg	Pre-tax extraordinary charges, reflecting non-recurring pre-tax charges.
ff_pe	Price-to-earnings ratio, a valuation metric comparing the company's market price to its earnings.
ff_int_exp_oth	Other interest expenses, not related to standard debt.
ff_com_eq_apic	Additional paid-in capital, representing the excess amount paid over the par value of stock.
ff_ebit_bef_unusual	Earnings before interest and taxes (EBIT), excluding unusual or non-recurring items.
ff_pe_dil	Diluted price-to-earnings ratio, accounting for potential dilution from convertible securities or options.
ff_int_exp_debt	Interest expenses related to debt obligations.
ff_entrpr_val	Enterprise value, representing the company's total market value, including debt and equity.

Table A.7: A Set of Key Momentum Drivers Found for the Finance Sector. The result of the comprehensive feature selection process.

A.8 Feature Importance Graphs

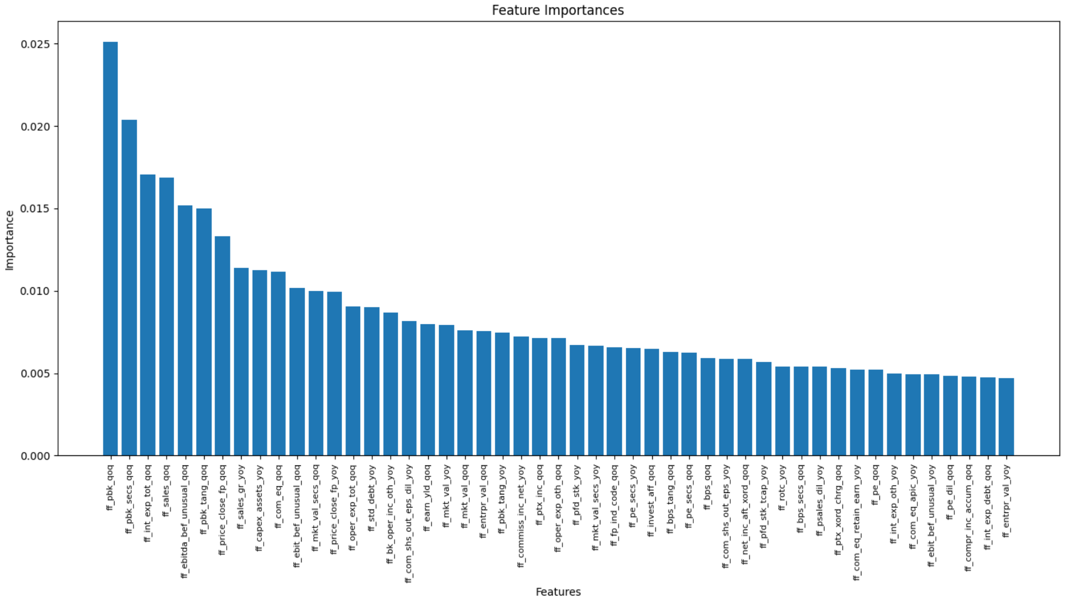


Figure A.1: Top Quintile of Features Selected from RF Regression Against Returns

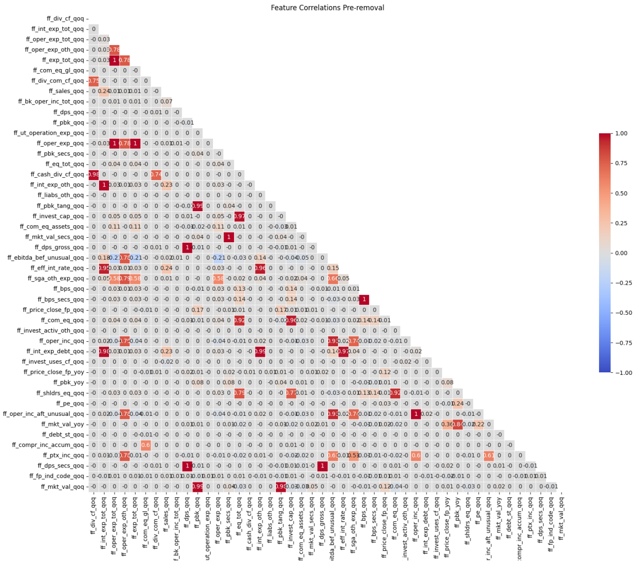


Figure A.2: Correlation Matrix Before Correlated Feature Removal

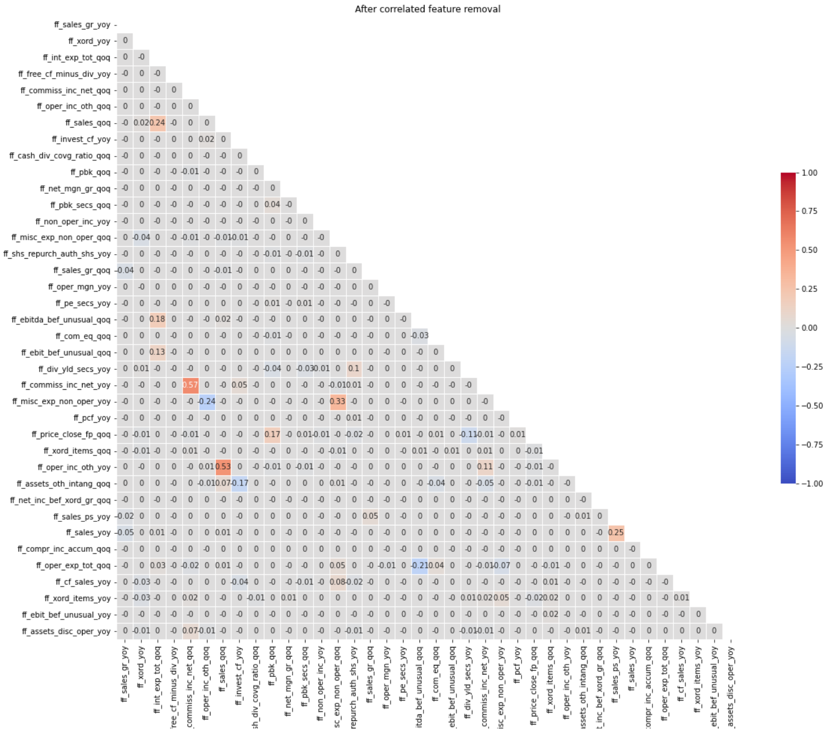


Figure A.3: Correlation Matrix After Correlated Feature Removal

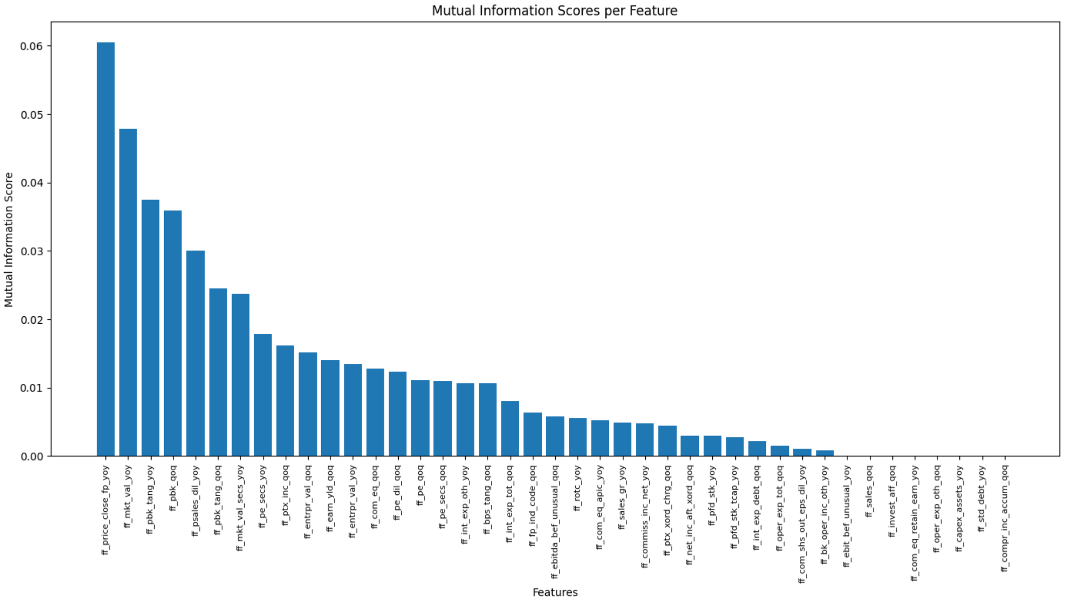


Figure A.4: Aftermath of Correlation Analysis: MI Shared with Returns for Remaining Features

Appendix B

Code and Visualisations - Max-Min Windowing Algorithm

B.1 Code for Labelling Momentum Turning Points in Price Data

```
def label_prices(df, column, split_window=40):
    roll_win = Window.partitionBy('fsym_id').orderBy('p_date').rowsBetween(-10, 0)
    win = Window.partitionBy('fsym_id').orderBy('p_date')

    df = df.withColumn("smooth_price", F.avg(col(column)).over(roll_win))

    df = df.withColumn("row_num", F.row_number().over(win))

    df = df.withColumn("Boundary", F.when((F.col("row_num") - 1) % split_window == 0, 1)
                                    .when(F.col("row_num") % split_window == 0, 1)
                                    .otherwise(0))

    df = df.withColumn("split_group", F.floor((F.col("row_num") - 1) / split_window))

    min_price = F.min(col("smooth_price")).over(Window.partitionBy('fsym_id', 'split_group'))
    max_price = F.max(col("smooth_price")).over(Window.partitionBy('fsym_id', 'split_group'))

    df = df.withColumn("Label", F.when(F.col("smooth_price") == min_price, 'min')
                                    .when(F.col("smooth_price") == max_price, 'max')
                                    .otherwise('regular'))

    boundary_df = df.filter(F.col("Boundary") == 1)

    boundary_df = boundary_df.withColumn("prev_date", F.lag(F.col("p_date")).over(win))
    boundary_df = boundary_df.withColumn("next_date", F.lead(F.col("p_date")).over(win))
    boundary_df = boundary_df.withColumn("prev_label", F.lag(F.col("Label")).over(win))
    boundary_df = boundary_df.withColumn("next_label", F.lead(F.col("Label")).over(win))

    boundary_df = boundary_df.withColumn("adjusted_label", F.when(
        (F.col("Label") == 'min') & (F.col("next_label") == 'max') & (F.datediff(F.col("next_date"), F.col("p_date")) <= 3), 'regular')
        .when(
            (F.col("Label") == 'max') & (F.col("next_label") == 'min') & (F.datediff(F.col("next_date"), F.col("p_date")) <= 3), 'regular')
        .when(
            (F.col("Label") == 'min') & (F.col("prev_label") == 'max') & (F.datediff(F.col("p_date"), F.col("prev_date")) <= 3), 'regular')
        .when(
            (F.col("Label") == 'max') & (F.col("prev_label") == 'min') & (F.datediff(F.col("p_date"), F.col("prev_date")) <= 3), 'regular')
        .otherwise(F.col("Label")))

    df = df.join(boundary_df.select("fsym_id", "p_date", "adjusted_label"), on=["fsym_id", "p_date"], how="left")

    df = df.withColumn("Label", F.coalesce(F.col("adjusted_label"), F.col("Label")))

    df = df.drop("row_num", "split_group", "prev_date", "next_date", "prev_label", "next_label", "adjusted_label")

    boundary_minmax_df = df.filter((F.col("Boundary") == 1) & ((F.col("Label") == "min") | (F.col("Label") == "max")))

    boundary_minmax_df = boundary_minmax_df.withColumn("prev_label", F.lag(F.col("Label")).over(win))
    boundary_minmax_df = boundary_minmax_df.withColumn("next_label", F.lead(F.col("Label")).over(win))
    boundary_minmax_df = boundary_minmax_df.withColumn("prev_price", F.lag(F.col("smooth_price")).over(win))
    boundary_minmax_df = boundary_minmax_df.withColumn("next_price", F.lead(F.col("smooth_price")).over(win))
    boundary_minmax_df = boundary_minmax_df.withColumn("prev_date", F.lag(F.col("p_date")).over(win))
    boundary_minmax_df = boundary_minmax_df.withColumn("next_date", F.lead(F.col("p_date")).over(win))

    boundary_minmax_df = boundary_minmax_df.withColumn("adjusted_label", F.when(
        (F.col("Label") == 'min') & (F.col("next_label") == 'min') & (F.datediff(F.col("next_date"), F.col("p_date")) <= 3) & (F.col("smooth_price") > F.col("next_price")), 'regular')
        .when(
            (F.col("Label") == 'max') & (F.col("next_label") == 'max') & (F.datediff(F.col("next_date"), F.col("p_date")) <= 3) & (F.col("smooth_price") < F.col("next_price")), 'regular')
        .when(
            (F.col("Label") == 'min') & (F.col("prev_label") == 'min') & (F.datediff(F.col("p_date"), F.col("prev_date")) <= 3) & (F.col("smooth_price") > F.col("prev_price")), 'regular')
        .when(
            (F.col("Label") == 'max') & (F.col("prev_label") == 'max') & (F.datediff(F.col("p_date"), F.col("prev_date")) <= 3) & (F.col("smooth_price") < F.col("prev_price")), 'regular')
        .otherwise(F.col("Label")))

    df = df.join(boundary_minmax_df.select("fsym_id", "p_date", "adjusted_label"), on=["fsym_id", "p_date"], how="left")

    df = df.withColumn("Label", F.coalesce(F.col("adjusted_label"), F.col("Label")))

    df = df.drop("prev_label", "next_label", "prev_price", "next_price", "prev_date", "next_date", "adjusted_label")

    minmax_df = df.filter((F.col("Label") == "min") | (F.col("Label") == "max"))

    minmax_df = minmax_df.withColumn("prev_label", F.lag(F.col("Label")).over(win))
    minmax_df = minmax_df.withColumn("next_label", F.lead(F.col("Label")).over(win))

    minmax_df = minmax_df.withColumn("adjusted_label", F.when(
        (F.col("Label") == 'min') & (F.col("prev_label") == 'min') & (F.col("Boundary") == 1), 'regular')
        .when(
            (F.col("Label") == ’max’) & (F.col("prev_label") == ’max’) & (F.col("Boundary") == 1), ’regular’)
        .when(
            (F.col("Label") == ’min’) & (F.col("next_label") == ’min’) & (F.col("Boundary") == 1), ’regular’)
        .when(
            (F.col("Label") == ’max’) & (F.col("next_label") == ’max’) & (F.col("Boundary") == 1), ’regular’)
        .otherwise(F.col("Label")))

    df = df.join(minmax_df.select("fsym_id", "p_date", "adjusted_label"), on=["fsym_id", "p_date"], how="left")

    df = df.withColumn("Label", F.coalesce(F.col("adjusted_label"), F.col("Label")))
    df = df.drop("prev_label", "next_label", "adjusted_label")

    minmax_df = df.filter((F.col("Label") == "min") | (F.col("Label") == "max"))

    minmax_df = minmax_df.withColumn("prev_price", F.lag(F.col("smooth_price")).over(win))
    minmax_df = minmax_df.withColumn("next_price", F.lead(F.col("smooth_price")).over(win))
    minmax_df = minmax_df.withColumn("prev_label", F.lag(F.col("Label")).over(win))
    minmax_df = minmax_df.withColumn("next_label", F.lead(F.col("Label")).over(win))
    minmax_df = minmax_df.withColumn("prev_date", F.lag(F.col("p_date")).over(win))
    minmax_df = minmax_df.withColumn("next_date", F.lead(F.col("p_date")).over(win))

    minmax_df = minmax_df.withColumn("adjusted_label", F.when(
        (F.col("Label") == ’min’) & (F.col("Label") == ’min’) & (F.col("prev_label") == ’min’) & (F.abs(F.col("smooth_price") - F.col("prev_price")) / F.col("prev_price") < 0.05) & (F.datediff(F.col("p_date"), F.col("prev_date")) <= 30), ’regular’)
        .when(
            (F.col("Label") == ’min’) & (F.col("next_label") == ’min’) & (F.abs(F.col("smooth_price") - F.col("next_price")) / F.col("next_price") < 0.05) & (F.datediff(F.col("next_date"), F.col("p_date")) <= 30), ’regular’)
        .when(
            (F.col("Label") == ’max’) & (F.col("prev_label") == ’max’) & (F.abs(F.col("smooth_price") - F.col("prev_price")) / F.col("prev_price") < 0.05) & (F.datediff(F.col("p_date"), F.col("
```

```python
                prev_date")) <= 30), 'regular')
        .when(
            (F.col("Label") == 'max') & (F.col("next_label") == 'max') & (F
                .abs(F.col("smooth_price") - F.col("next_price")) / F.col("
                next_price") < 0.05) & (F.datediff(F.col("next_date"), F.
                col("p_date")) <= 30), 'regular')
        .otherwise(F.col("Label")))

    # Join adjusted labels back to the original dataframe
    df = df.join(minmax_df.select("fsym_id", "p_date", "adjusted_label"),
        on=["fsym_id", "p_date"], how="left")

    # Update the labels in the original dataframe
    df = df.withColumn("Label", F.coalesce(F.col("adjusted_label"), F.col("
        Label")))

    df = df.drop("prev_label", "next_label", "prev_price", "next_price", "
        prev_date", "next_date", "adjusted_label")

    return df
```

B.2 Momentum Turning Point Visualisations for Max-Min Windowing Approach

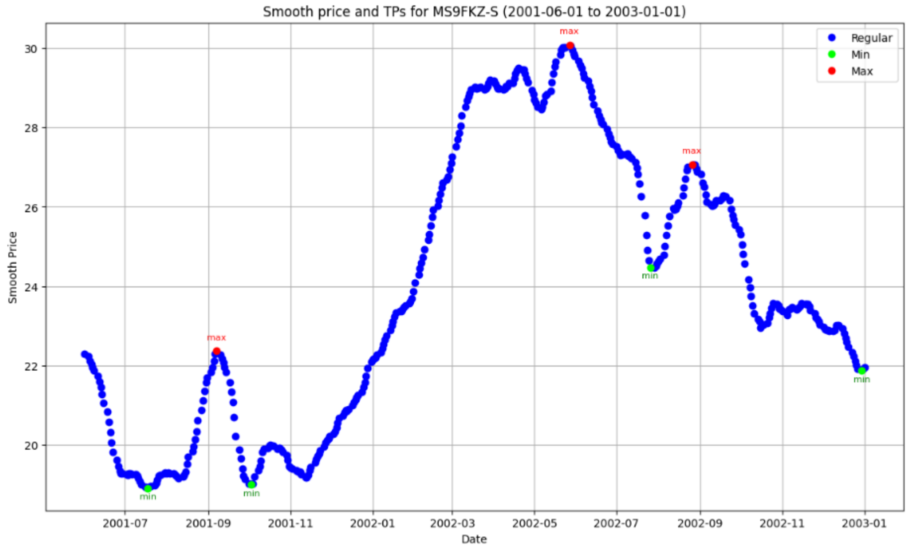


Figure B.1: Daily Turning Point Labels for MS9FKZ-S

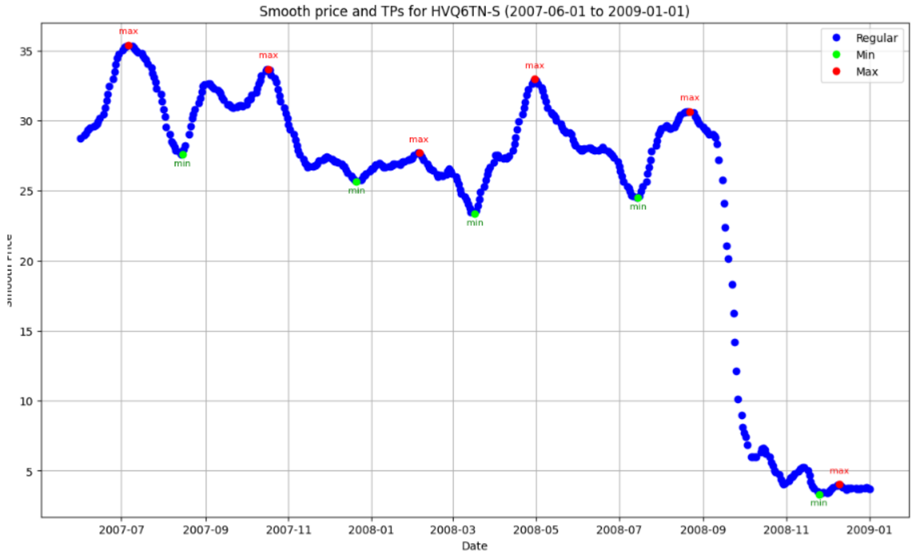


Figure B.2: Daily Turning Point Labels for HVQ6TN-S

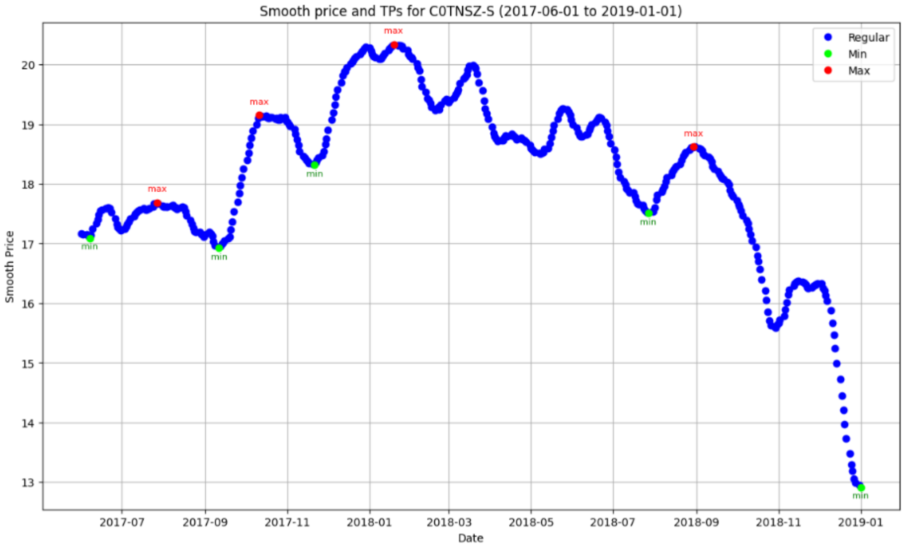


Figure B.3: Daily Turning Point Labels for C0TNSZ-S

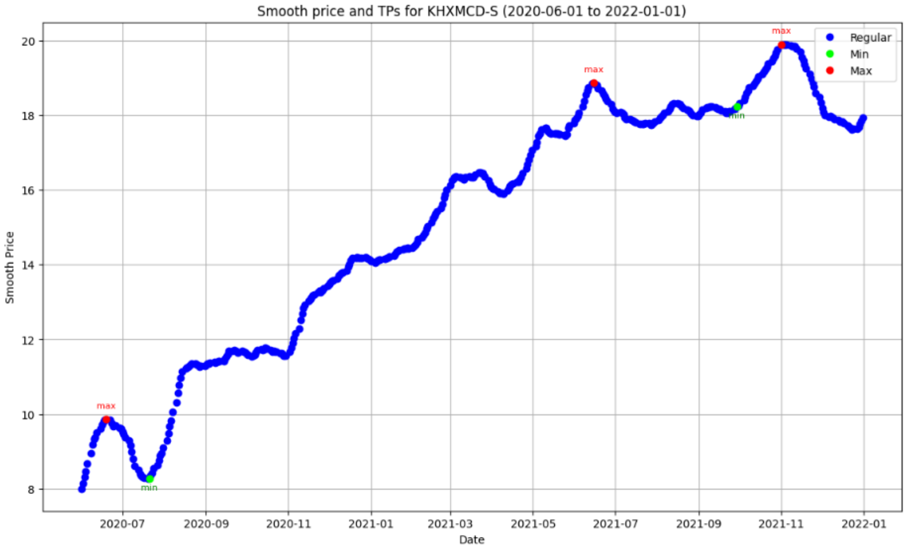


Figure B.4: Daily Turning Point Labels for KHXMCD-S